NASA TECHNICAL NOTE

NASA TN D-8240

NASA TN D-8240

CLASSIFICATION SOFTWARE TECHNIQUE ASSESSMENT

Robert R. Jayroe, Jr., Robert Atkinson,
B. V. Dasarathy, Matt Lybanon,
and H. K. Ramapryian

George C. Marshall Space Flight Center
Marshall Space Flight Center, Ala. 35812

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MAY 1976

TECHNICA PAGE

1. REPORT NO.
TN D-8240

2. GOVERNMENT ACCESSION NO.

4 TITLE AND SUBTITLE

Classification Software Technique Assessment

5. REPORT DATE
May 1976

6. PERFORMING ORGANIZATION CODE
M-172

7. AUTHOR(S) Robert R. Jayroe, Jr., Robert Atkinson,
B. V. Dasarathy, Matt Lybanon, H. K. Ramapryian

8. PERFORMING ORGANIZATION REPORT #

9. PERFORMING ORGANIZATION NAME AND ADDRESS

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

10. WORK UNIT NO.

11. CONTRACT OR GRANT NO.

13. TYPE OF REPORT & PERIOD COVERED

Technical Note

12. SPONSORING AGENCY NAME AND ADDRESS

National Aeronautics and Space Administration
Washington, D. C. 20546

14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

This work was sponsored by the Office of Applications' Data Management Program. Prepared by Data Systems Laboratory, Science and Engineering Directorate, MSFC

16. ABSTRACT

The need for this work was generated from two main sources. One is an increasing interest on the part of local user communities to obtain software for analysing remotely sensed multispectral imagery and/or to obtain the results of that analysis. Secondly, because of the sheer volume and complexity of the data, a definite processing problem exists. In addition, many different software techniques and variations of techniques have been developed for the analysis of such data. As a result, the requirement exists to determine the status of the image analysis state-of-the-art, identify the problem areas, and find solutions so that the processing problem can be relieved and the state-of-the-art advanced. Determining the status of the state-of-the-art is a statistical problem that is due to the wide variation in image data sets themselves as well as to the variety of applications for which the data are analyzed.

Therefore, from a user's point of view, this report describes the resources that are required to utilize a particular software program, how the program analyzes the data, and the performance of that program for an application and data set that was provided by a user. The report is also an initial effort at establishing a statistical performance base, for various software programs with regard to different data sets and analysis applications, to determine the status of the state-of-the-art.

17. KEY WORDS

Classification, Remote Sensing, Classification Technique Evaluation

18. DISTRIBUTION STATEMENT

Categories 43 and 61

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	153	$6.25

For sale by the National Technical Information Service, Springfield, Virginia 22161

ACKNOWLEDGEMENTS

The authors wish to thank Dr. C. T. N. Paludan of the Earth Resources Office, Marshall Space Flight Center for coordinating this effort with the Tennessee State Planning Office of Nashville, Tennessee, and to thank Mr. John Wilson and his staff at the Tennessee State Planning Office for their cooperation and participation in the conduct of this assessment. The authors also wish to acknowledge Dr. A. Bond of the Computer Science Corporation, Huntsville, Alabama, for his helpful suggestions during the evaluation and for editing and coordinating the writing of this report.

FOREWORD

This work was directed by Dr. R. R. Jayroe of the Information Sciences Division, Data Systems Laboratory, Marshall Space Flight Center, and was accomplished by the efforts of Dr. Jayroe and Dr. Bob Atkinson, Dr. B. V. Dasarathy, Mr. Matt Lybanon, and Dr. H. K. Ramapryian of the Information Systems Section, Computer Science Corporation, Huntsville, Alabama, under Contract NAS8-31639.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Cont'd.)

LIST OF ILLUSTRATIONS

Figure | Page

Section I — Classification Software Assessment — Requirements and Approach

Section II — Program Descriptions

Section III — Technique Assessment

LIST OF ILLUSTRATIONS (Cont'd.)

Figure Page

Section III — Technique Assessment (Cont'd.)

LIST OF TABLES

Table Page

Section I — Classification Software Assessment — Requirements and Approach

Section II — Program Descriptions

Section III — Technique Assessment

LIST OF TABLES (Cont'd.)

SECTION I

CLASSIFICATION SOFTWARE ASSESSMENT – REQUIREMENTS AND APPROACH

1.0 INTRODUCTION

One of the roles of the MSFC Discipline Center for Data and Information Management is to provide objective and impartial evaluation of data system techniques and developments. Currently many different software techniques are finding application in the extraction of information from earth resources imaging systems, but few guidelines exist to direct a concerned user to the technique most suitable for his application. As a partial fulfillment of this role and as a result of increasing user interest in obtaining software techniques and analyses, this document is mainly concerned with the evaluation development in the area of pattern classification software. The purposes of this document are:

- to document, in laymen's terms whenever possible, the resources required to utilize a particular technique, what the technique does, and how well the technique performs,
- to establish a standardized procedure for evaluating classification techniques that could be extended to other technique development, and
- to determine selected areas of emphasis for future technique development, as well as minimize duplication of effort.

In order to be an impartial evaluator, MSFC has de-emphasized local technique development and therefore has no stake in any particular technique. In addition, another OA Program area, the Earth Resources Office of the Data Systems Laboratory, MSFC, was used as an intermediary between the evaluation of results and the performance of the analysis. This procedure produced an air of anonymity to the technique and personnel that provided the results, which in turn provided a means of obtaining unbiased and critical user evaluation. The work arrangement was organized such that the Information Management and Analysis Branch and Computer Sciences Corporation provided the analysis results and performed the computer related evaluation, the Earth Resources Office provided the data and user, and the user provided a set of requirements, ground truth, and an evaluation of results.

The methodology for conducting technique assessment experiments and for establishing the standardized evaluation procedure is illustrated in Figure 1-1. User specified data sets are classified by each technique in turn, and the classification performance is observed, using the supplied verification data (ground-truth map) for accuracy assessment. The results of this work, classification maps and other performance data, are transmitted to the end

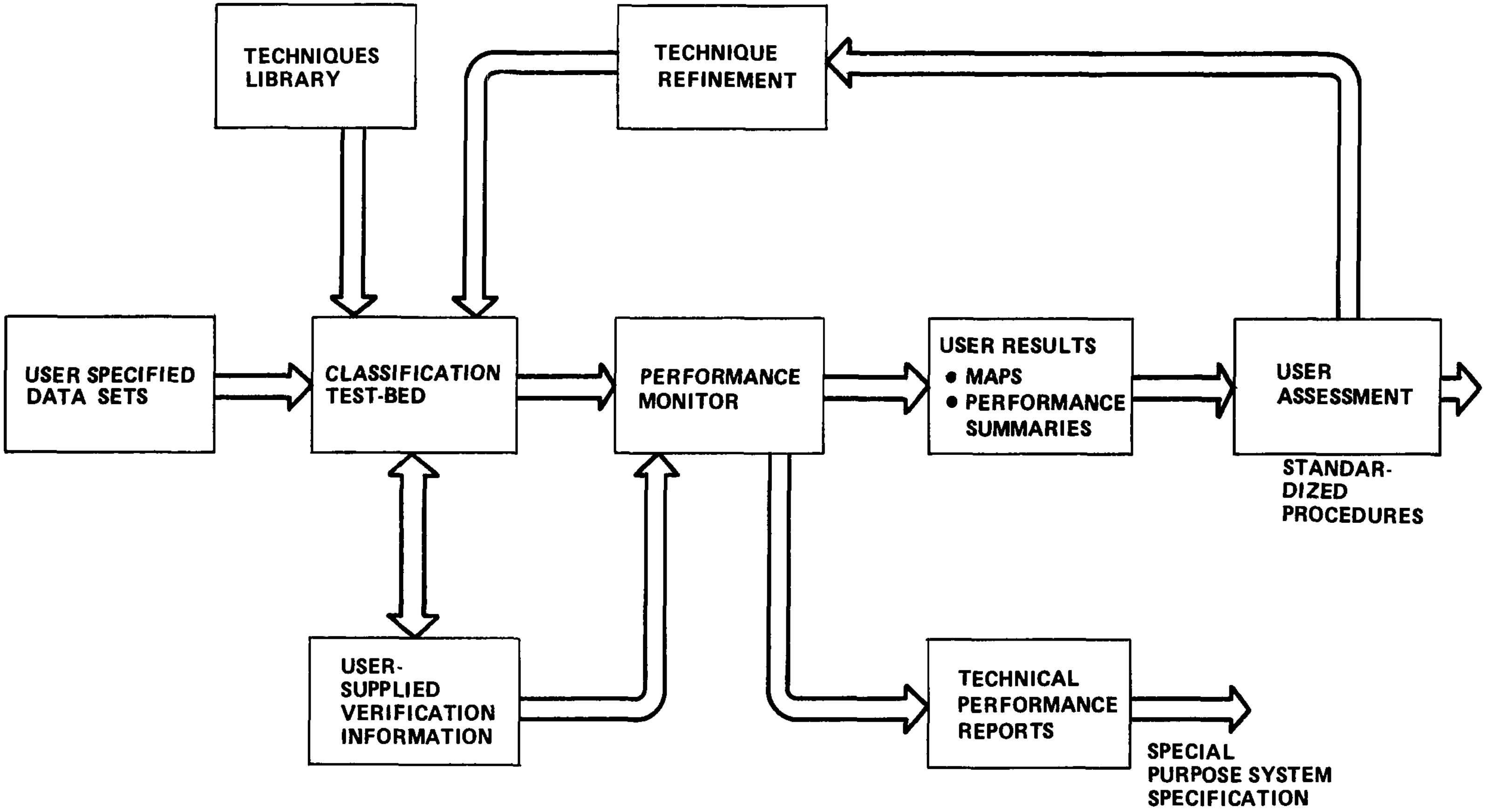


Figure 1-1. Methodology for Classification Technique Assessment

user, who evaluates their efficacy within the context of his operational environment and intended application. Currently, the computations are carried out in a batch mode, and the technique results are transmitted to the end user in the form of hard copy. In the future, an interactive implementation of the classification testbed will provide a mechanism for the user to interact with each technique and to assess results instantly through the medium of video display screens. Once the user's assessment has been made, those techniques whose results can be directly assimilated, in the form of data products, within the user's application can be considered as candidates for standardization and for wide dissemination among the user community. Those techniques whose results exhibit shortcomings or weaknesses are analyzed, refined to accommodate the user assessment, and evaluated again. Also, those techniques which emerge as having the potential for high operational utilization, are analyzed to determine efficient mechanisms, or special purpose devices, to simplify and reduce the cost of their application.

The initial technique selection was based upon those techniques that were most heavily promoted, recommended, used, and readily available; but in order to broaden the base of the evaluation, other techniques will be included later, as well as other data sets, users, and requirements. The techniques assessed and reported in this first document release are itemized in Section II.1. In addition, the technique evaluation was mainly conducted on an IBM 360/65, but the utilization or mention of any computer or related hardware should not be construed as an endorsement of a commercial product.

Suggestions from users or readers of this document of other techniques that could usefully be included in this assessment program will be welcomed.

2.0 BACKGROUND: CLASSIFICATION/PATTERN RECOGNITION TECHNIQUES

In the context of this report, classification techniques are discussed in terms of applying the techniques to image analysis, where the imagery is in a digital form. The analysis could include such application areas as character recognition, diagnosis of blood cells, recognition of chromosomes, X-ray image analysis, fingerprint classification, and natural resource recognition, to name a few. In all these cases, the goal of the user is to obtain, in some form, a "classification" of the observations relevant to his applications. From the point of view of a user, a pattern recognition system can be represented as shown in Figure 2-1. The "black-box" in this figure can be filled in several ways. While the user is not necessarily interested in unlocking the locks of the black-box, he would certainly like to find out how well the available alternatives would satisfy his needs. For instance, in processing the large quantities of data generated by spaceborne sensors, the user might wish to have a system which would assure high speed commensurate with the rate of observation, whereas in other environments he might be willing to sacrifice processing speed in favor of higher accuracy.

The main concerns of a user of a pattern recognition system (PRS) are how much he should spend for the system and what he will get out of it. The complexity of answering these apparently simple questions is revealed by a slight peek into the black-box.* The system can be subdivided as shown in Figure 2-2. The measurement process is added outside the PRS as a separate box since the user may not be concerned with selecting a measurement process, even though the design or choice of the PRS might depend on it. The design process for a PRS is illustrated in Figure 2-3.

Each of the blocks in the design process is seen to receive data from the others. Pattern analysis consists of using the knowledge of the problem at hand to direct the measurement process and subjecting the data to a variety of tests to identify structures, if any, present in the data that may lead to better feature definition and classification. The word "feature" refers to an entity that might be derived from some initial measurements. Feature extraction is the process by which a measurement vector is transformed into a new vector, the goal being to find features that are effective in discriminating between pattern classes. A classifier uses the features and assigns the observations to various classes. A large number of choices is available for each of the steps in the design process.

In order to perform a reasonable evaluation of the classification algorithms, it is necessary to be sufficiently exhaustive in enumerating the currently existing algorithms and keep the list up to date as new techniques are developed. Since

*The development of Figures 2-2 and 2-3 follows that of 'Patterns in Pattern Recognition: 1968-1974," Kanal, L., IEEE Trans. on Information Theory, Vol. IT-20, No. 6, November 1974.

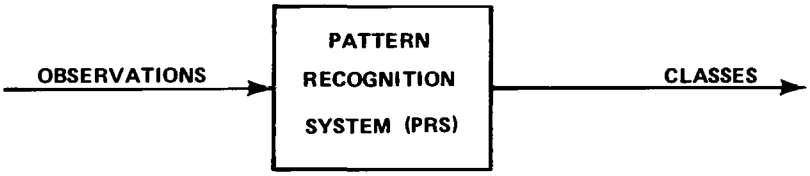


Figure 2-1. A User's View of a Pattern Recognition System

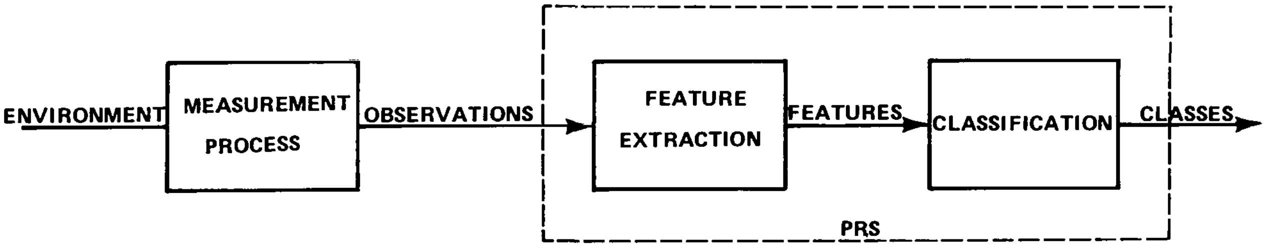


Figure 2-2. A Slightly More Detailed View of the Pattern Recognition System

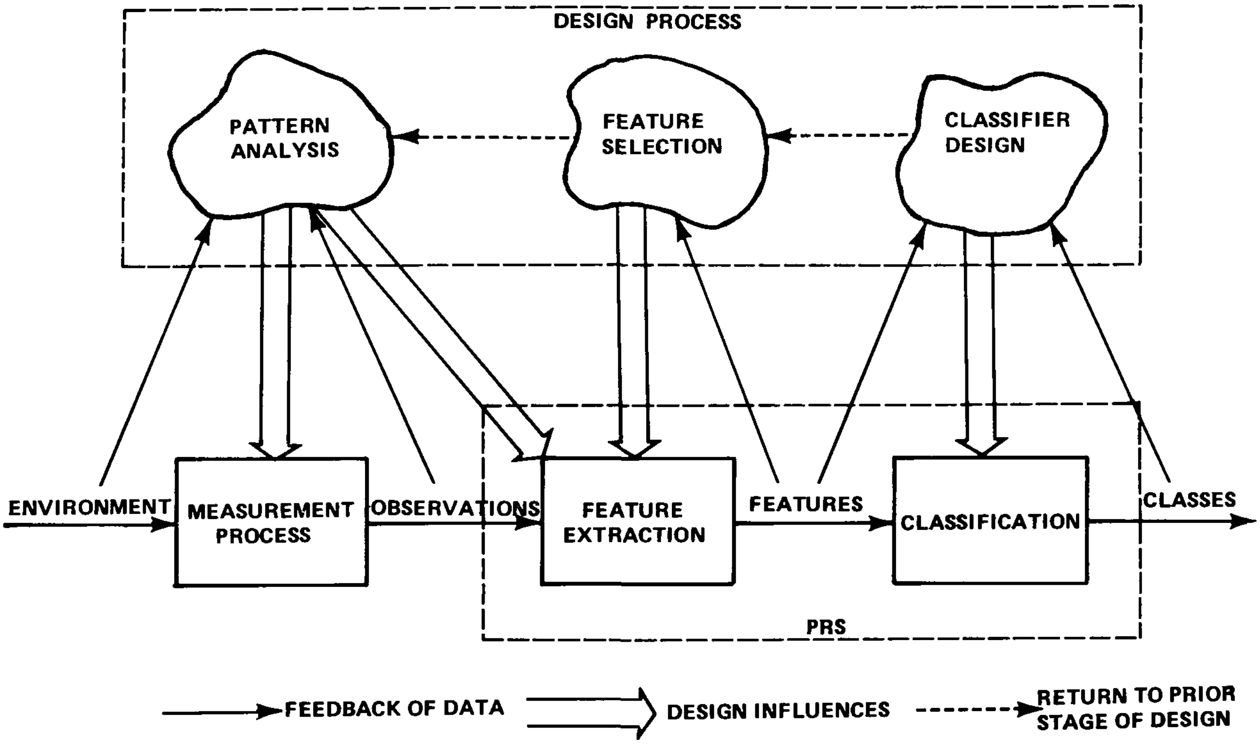


Figure 2-3. Design of a Pattern Recognition System

the number of techniques developed is quite large, it is convenient to categorize them as follows, based on the knowledge that the user has about the input data.

As mentioned earlier, the input to a pattern recognition system is a sequence of observations, which may be called "measurement vectors." The user might have varying degrees of knowledge about the measurements. A simple example to illustrate this is the case of remotely sensed data of a region on earth (say, LANDSAT data). Suppose the goal of the user is to find the land use categories in a given area. He might, in some cases, know the categories he is looking for and the "ground truth" (i.e., the class designations) at a small subset of locations from the remotely sensed image. On the other hand, he might know nothing about the data and have to rely entirely on the classifier to analyze it and produce classifications. It is common, in the literature on pattern recognition, to look upon the classifier as a "student" and the information supplied by the user about the ground truth as a "teacher." With this analogy, the classifier can be broken into two phases, the first called "learning" and the second called "classification" (see Figure 2-4). When the ground truth is known, the learning is said to be "supervised." When there is no knowledge of ground truth, the student is unaided by the teacher and hence the learning is said to be unsupervised.

There are several cases between these two extremes. The user might know, say, by photointerpretation of a high altitude aerial photograph, that a certain section of the region under study is representative of all the classes of interest, but may not be able to supply the class names corresponding to each point in the image. In this case we say that the "training samples" are known, but the "labels" for them are unknown. The classifier should first separate the classes present within the training set and, based on that experience, classify the entire data set. This can be called "pseudosupervised learning" (Figure 2-5).

Another case arises where the user knows the labels for the training samples, but is not certain of their correctness. In this case, the labels are said to be unreliable. This situation could occur when a low altitude photograph is used to obtain the ground truth information. Suppose several photointerpreters are used to derive the labels and their results differ. Then we can associate a confidence level (reliability or probability of being correct) with each of the labels, depending upon how many of the interpreters agreed on that label. (Equivalently, if we have a certain level of confidence in a particular photointerpreter based on his past record, we can associate a reliability measure with his labels.) In this case, the learning is said to be through an "imperfect teacher"*(see Figure 2-6).

*K. Shanmugam, "A parametric procedure for learning with an imperfect teacher," IEEE Trans. Inform. Theory. IT-18, pp. 300-302, March 1972.

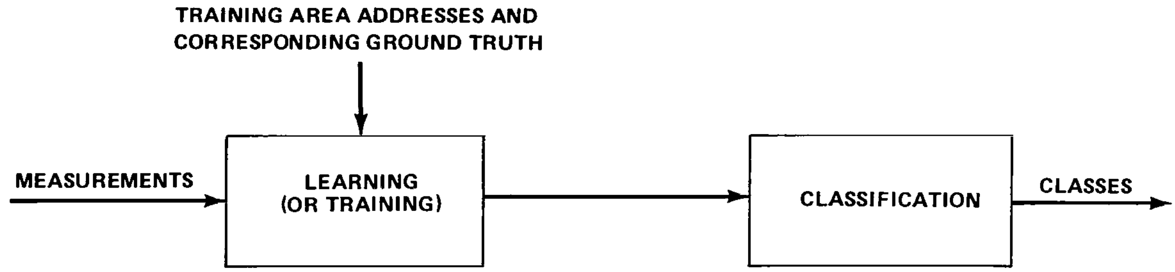


Figure 2-4. Supervised Learning and Classification

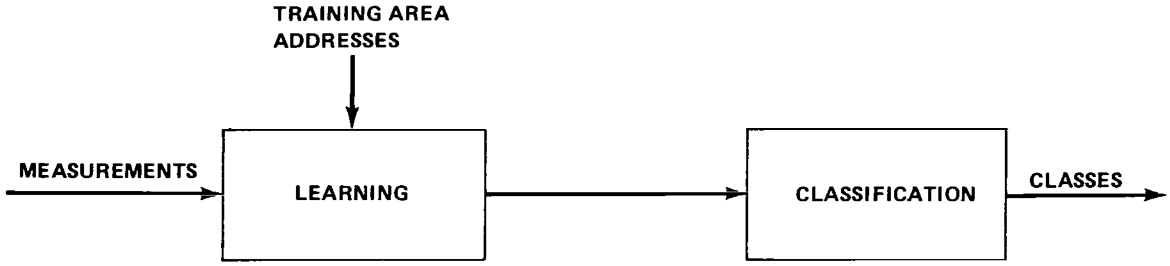


Figure 2-5. Pseudo-Supervised Learning and Classification

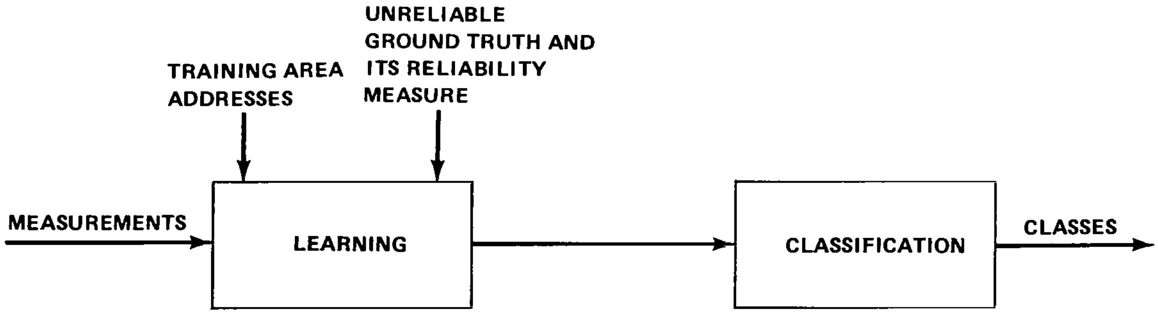


Figure 2-6. Learning Through an Imperfect Teacher

Consider further the case where the photointerpreter's pase record in unknown and the labels produced by him on a low altitude aerial photograph are used with the training samples. Then the labels are unreliable and their reliability measure is unknown. In this case, the learning is said to be through an unfamiliar teacher*(see Figure 2-7).

The categorization of classification methods considered so far is based on the degree of knowledge of the training samples. Another type of division is made depending upon the knowledge of the multivariate probability distribution for each class. If the distributions are completely known, then there is, indeed, no need for using any training samples. The distributions can be directly used by the classifier. (We can say that the learning phase in this case consists of converting the distributions into convenient "discriminant functions"--functions which are computed and then compared by the classifier to make the classification decision.) When the distributions are known only in functional form with a finite set of unknown parameters (or it is reasonable to assume so), the parameters should be determined on the basis of observed samples. This is called "parametric learning." Situations where even the functional form of the distributions are unknown call for "nonparametric learning."

Of the two categorizations described above, parametric and nonparametric, the former can be looked upon as based on the level of detailed knowledge of a small subset of the data, while the latter may be regarded as dependent on the degree of knowledge of the gross behavior of the observations. Figure 2-8 shows a flowchart indicating the choice of the types of classification methods based on the user's knowledge about the data to be classified. Table 2-1 summarizes the characteristics of some of the techniques that have been most widely used in the practical applications of pattern classification.

*B. V. Dasarathy and A. L. Lakshminarasimhan, "Sequential Learning Employing Unfamiliar Teacher Hypothesis (SLEUTH) witn Concurrent Estimation of Both the Parameters and Teacher Characteristics," Int. J1 of Computers and Information Sciences, Vol. 5, No. 1, January 1976.

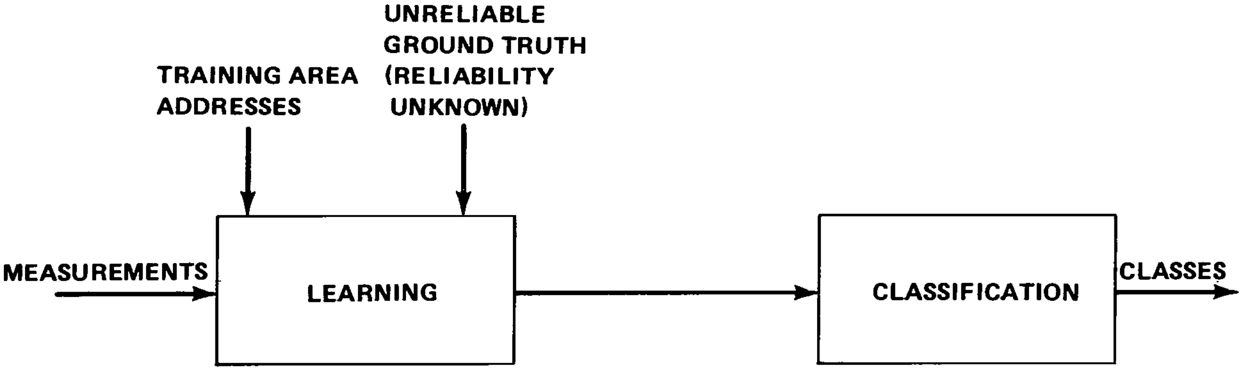


Figure 2-7. Learning Through an Unfamiliar Teacher

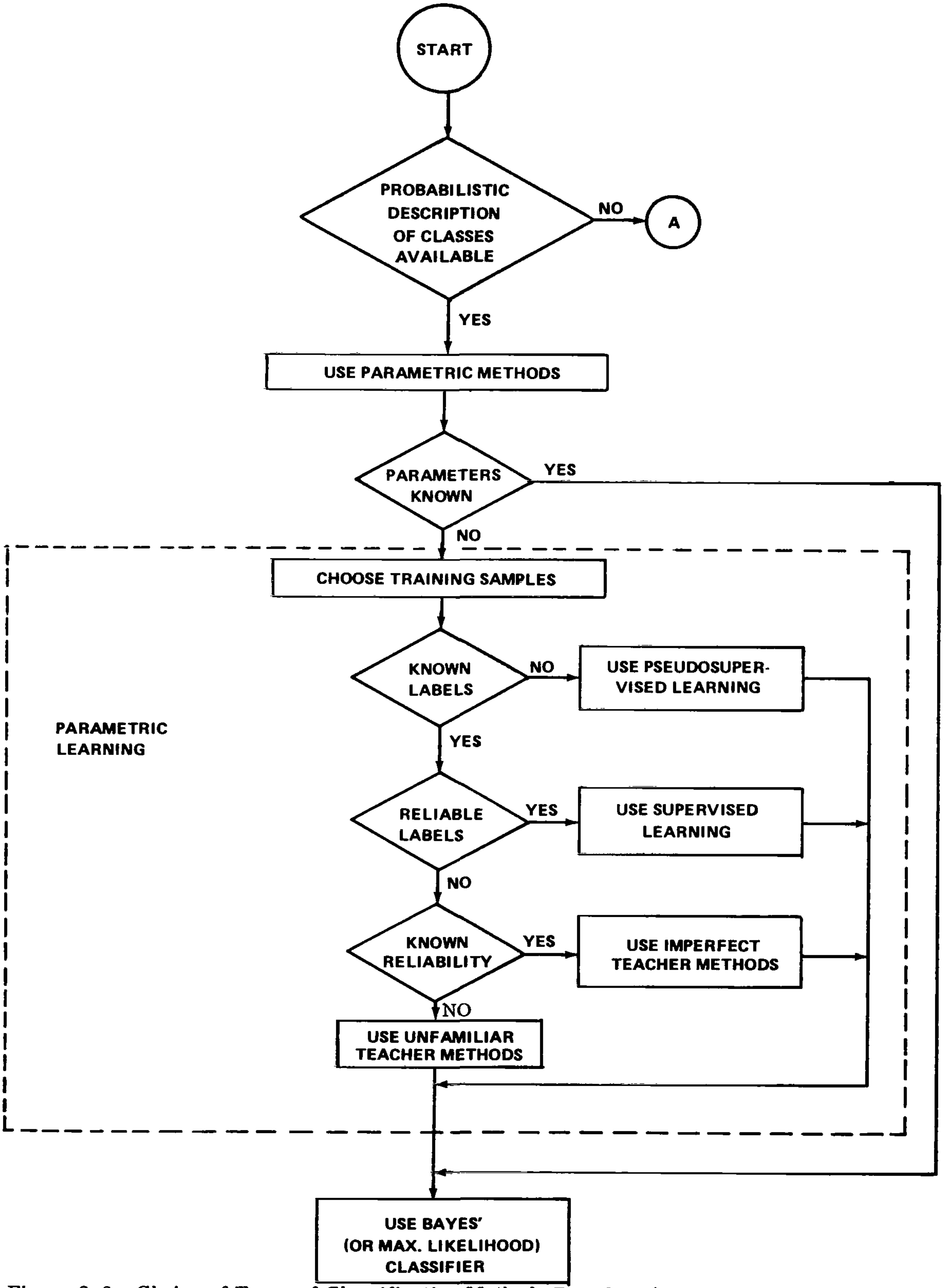


Figure 2-8. Choice of Types of Classification Methods Based on Amount of Input Information

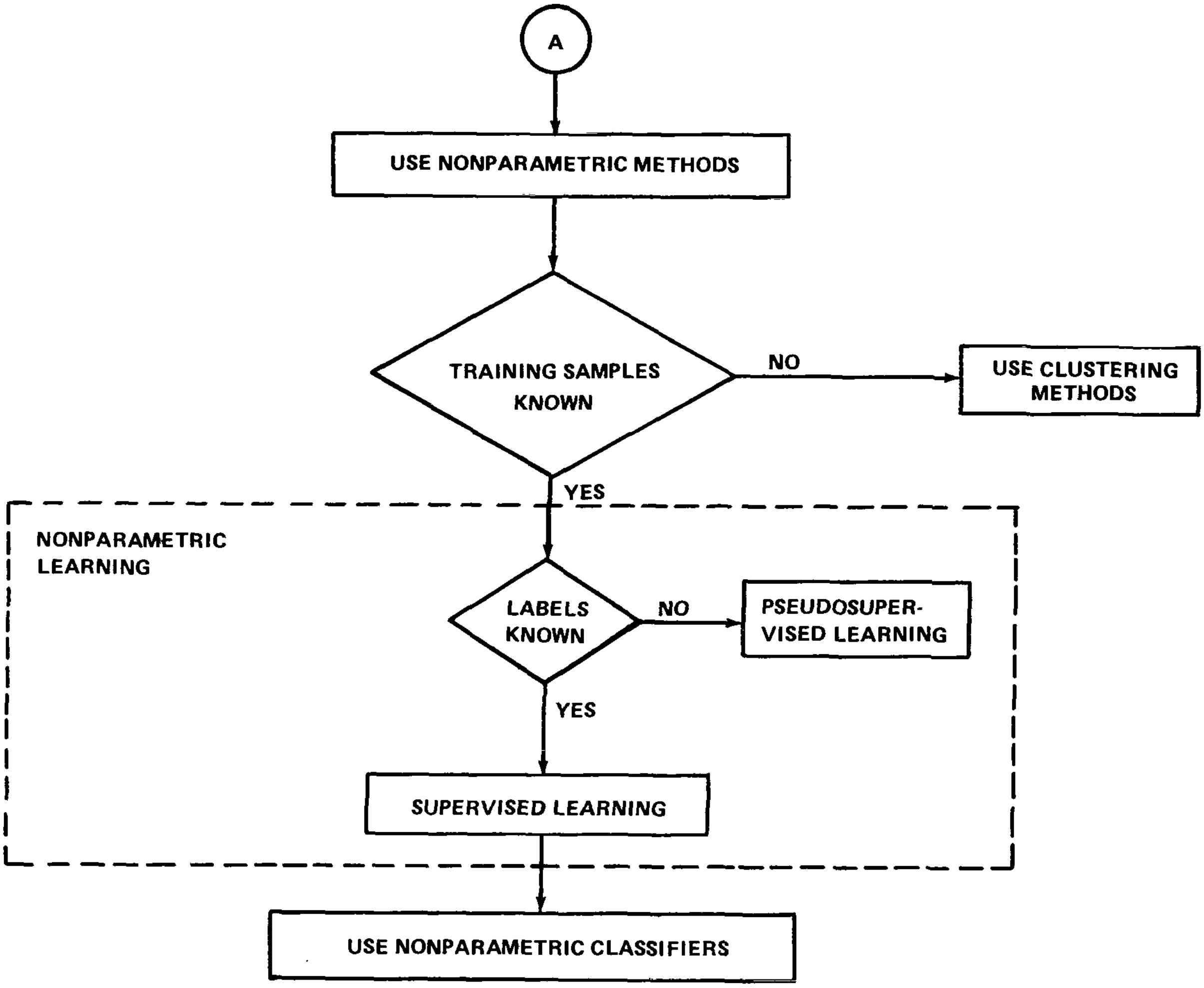


Figure 2-8. (Continued)

Table 2-1. A Partial List of Classification Methods

Classification Method	Categorization 1	Categorization 2	Comments
Bayes	Supervised	Parametric	Minimizes "average risk" of misclassification. Requires knowledge of a priori probabilities of occurrence of each class.
Maximum Likelihood	Supervised	Parametric	Minimizes average risk of misclassification when the probabilities of occurrence of each class are equal. When the conditional density functions are assumed Gaussian, this is a quadratic classifier used in LARSYS, ELLTAB, etc.
K-Nearest Neighbor	Supervised	Nonparametric	Finds class assignments of K-nearest neighbors and puts given samples in the majority class.
Prototype	Supervised	Nonparametric	Represents each class by a prototype and assigns a point to nearest prototype.
Linear	Supervised	Nonparametric	Linear classifier is a general term to encompass techniques which use linear surfaces (hyperplanes) to separate classes. There are several iterative methods for deriving such hyperplanes.
Piecewise Linear	Supervised	Nonparametric	This is a generalization of linear classifiers. Useful when the classes are not separable by hyperplanes (either pairwise or individually from all other classes).
Quadratic and Higher Order Polynomial	Supervised	Nonparametric	Use higher order surfaces for separating classes. The surfaces can be found using the same methods as for the Linear Classifier by suitable enlargement of the feature vectors.
Distance Based Clustering	Unsupervised	Nonparametric	There are several methods which use distance measures to group data into clusters. These are iterative methods and vary slightly from each other in the details of handling, initiation, and updating of clusters (e.g., K-Means, CURRY, SSCP, CLUSTER).
Density Based Clustering	Unsupervised	Parametric	Assuming form of probability density functions, find cluster assignments such that a measure of overlap is minimized.
Density Based Clustering	Unsupervised	Nonparametric	Approximate multivariate density by sample histograms or some other smooth functions and seek their local maxima (modes) (e.g., HINDU system).
Table Look-Up	-	-	Can be used to implement any decision rule obtained from any classification method. (ELLTAB is an example of its use for quadratic decision rules.)

3.0 WHY IS TECHNIQUE EVALUATION NECESSARY?

In recent years, there has been a great deal of interest in classification/pattern recognition techniques as evidenced by the hundreds of journal articles and books on the subject, with the main concentration of effort being on technique development. The large number of different approaches that have been incorporated into the classification technique development is mostly a direct result of the multidisciplinary nature of the applications and tends to indicate that no single approach is able to satisfy a large class of users.

Results evaluating different techniques have been published, but the majority of these efforts consists of the evaluation of a technique by the original developer. This tends to preclude the application of other known techniques, and emphasizes a particular application to the exclusion of other possible related applications. Also, the evaluation often is performed on only one data set, or even using only simulated data so that generalization to live applications may be questionable. In other cases, the developer may concentrate on particular aspects of a technique without regard to successful performance in an operational environment. Generally, the evaluations are performed on different computers, so that it is difficult to compare the operational characteristics of different techniques. As illustration of the over-emphasis of mathematical development in the field without regard for applications, a fair comment is a quotation from a book review.*

> "While he should not be blamed for the unsatisfactory state of the art, he can be blamed for not making any attempt to convey to the reader a sense of the effectiveness and ineffectiveness of his methods. There are almost no applications (of 242 pages, only 6 are concerned with actual pattern recognition experiments). Thus a new sacred cow of mathematical machinery is created--its priesthood will probably make a good academic living regardless of whether the cow gives any milk."

As a result, the evaluations that are available are difficult to piece together to obtain an overall visibility concerning technique development.

The major part of the problem of obtaining a comprehensive evaluation of classification technique development is that it is a formidable task and no one individual has the computer hardware resources or variety of users' interfaces. Because

*Bremmerman, H. J., "Review of 'An Introduction to Mathematical Techniques in Pattern Recognition' by H. C. Andrews," American Scientist, Vol. 62, pp. 244-245, 1974.

of this problem, there has been no overall coordinated attempt at standardization of

- evaluation criteria,
- unbiased evaluation procedures, and
- computer hardware and software programming practices,

as well as a generalization of applications to include

- multidisciplinary users and data sets and
- an end-to-end systems approach to the evaluation.

Since the resources and user interfaces already exist within NASA, the establishment of such standardization and generalization practices could provide the means to focus the technique development for

- identifying problem areas,
- emphasizing areas of needed development and minimizing duplication of effort, and
- optimizing technology transfer to users.

4.0 EVALUATION CRITERIA

There are probably many ways to evaluate classification techniques, but from a user's point of view the three most important areas of concern appear to be:

- the resources required to run the program and perform an analysis,
- a description of the analysis process, and
- the performance of the technique.

Ideally in a systematic evaluation, these factors would be addressed in both their quantitative and qualitative aspects, with the qualitative aspects rates subjectively on a scale of excellence so that realistic comparison of the techniques' various attributes may be made. At this point, however, insufficient testing has been performed to permit an adequate qualitative assessment of the different classification techniques. Later issues of this document will include such an assessment, and will also incorporate a summary of technique performance as a function of disciplinary application.

For the present, the evaluation criteria used stress the quantitative aspects, and are intended to provide a user with necessary information so that estimates may be made of initial cost of resources (if the purchase of a system is being considered), of operating costs (if the technology is being transferred to an existing system), and of potential cost/benefit ratios. In addition, with regard to the development of the techniques themselves, the evaluation criteria are designed to provide some standardization in relative comparison of different classification techniques and their performance, with a view to identifying areas where improvements, in concept or in implementation, are necessary.

Section II of this document presents descriptions of each classification technique that has been evaluated. The material is organized to address the Resource Requirements, Analysis Process, and Performance Characteristics of each technique, as described in more detail below. For convenience of the reader in cross correlating the performance of the various techniques, Section III reports the results of evaluation tests on a variety of data sets.

4.1 RESOURCE REQUIREMENTS

The quantitative aspects of the resource requirements are essentially concerned with the computer hardware necessary to run the program. These resources consist of:

- number of tape drives,

- amount of mass storage,
- amount of core memory,
- number of routines that are specialized to a particular computer or hardware component,
- number and type of output devices, and
- number of input variables required to run the program.

The qualitative aspects of the resource evaluation consist mainly of knowledge available to the user about the program and the data set to be analyzed. Under the first category, there is a subjective evaluation concerning the adequacy of the technique documentation, which preferably should also include recent reports on applications or demonstrations of the technique if available, and a discussion of the types of input data that can be utilized, as well as input/output formats. The second category concerns the user's knowledge of the data set. What the user wants to do with data will to a large extent determine the type of technique (unsupervised or supervised) to be used on the data set. This in turn will impose a requirement for certain types of expertise and experience, for example in photointerpretation and in statistical data analysis.

4.2 ANALYSIS PROCESS

There are two basic types of classification techniques, supervised and unsupervised, each of which may be broadly subdivided into two categories, parametric and nonparametric. Generally, supervised techniques are used when ground truth is available to classify the entire data set or to identify a few specific classes. Unsupervised techniques are generally used when ground truth is not available to classify the entire data set. If either of the approaches assumes a model for the distribution of the data, the technique is called parametric, as opposed to nonparametric.

The purpose of this subsection in the technique descriptions is to provide a layman with a reasonable understanding of the classification analysis process and the role that a user plays in the analysis, as well as to contrast differences and highlight similarities between the various techniques. The description, in conjunction with the performance characteristics, leads to a categorization of classification techniques for indicating problem areas, tradeoffs between the various techniques, and needed research in technique development. This subsection also contains a list of input parameters with definition and uses.

4.3 PERFORMANCE CHARACTERISTICS

The performance characteristics are intended to indicate operational costs, cost/benefits, and maximum capabilities of the various techniques. Those quantities that can be enumerated are:

- computer time,
- relative accuracy in terms of direct pixel comparison of ground truth data and classification maps,
- maximum number of channels,
- maximum data set size,
- maximum number of clusters or classes,
- cost/benefit estimates in terms of relative accuracy and the use of conventional techniques, and
- manhours required by the user in the analysis.

The last two items tend to be subjective, since they depend on the type and quality of ground truth as well as on the human capabilities applied to the successive phases of photointerpretation, classifier training, and iteration a number of times through the analysis process to attain satisfactory results.

The qualitative aspects of the performance characteristics consist of:

- a description of all output information products,
- sensitivity of results to input parameter values,
- sensitivity of results to starting conditions,
- sensitivity of results to factors peculiar to a particular application or data type, and
- quality control of results, which include discussions on restart capabilities, modularity of program, iteration dependence, and risk of not obtaining the desired results at an early stage in the analysis.

4.4 CONDUCT OF USER EVALUATION

In order to obtain a complete evaluation of a technique, it is desirable to utilize a variety of data sets and work with users in a multiplicity of disciplines. Also, in order to both benefit the technique development and satisfy user requirements, it is desirable to have the user provide ground truth maps for pixel-by-pixel accuracy comparison and participate in the evaluation. The user can provide important feedback by examining the data products and providing written comments to questions such as those listed below.

- Where were we able or unable to satisfy your requirements in providing computer-derived products ?
- Which computer technique best satisfies your requirements and how would you rank them in order of satisfaction ?
- What do you consider to be the shortcomings and good points of the results produced by each technique ?
- What are the criteria by which you judge the results ?
- What improvements or changes need to be made in the analysis procedure or output product ?
- Which techniques provide information beyond what is contained in the ground truth maps that is useful or improves the accuracy ? If so, what new information was provided or where was the improvement noted ?
- What is the cost/benefit, if any, you would derive from using computer versus conventional techniques ?
- What would be your opinion on using the best technique at your facility and at your cost to produce computer products for public use ?

In order to maintain a high degree of objectivity in the evaluation, it is desirable not to have a technique developer utilize his own technique to provide results, but instead use an independent agent or middleman to provides results to a user.

5.0 PROCEDURES

In conducting a systematic assessment of classification techniques, certain procedures must be adopted to achieve consistency in results and to assure that relative comparisons have meaning. It is most important that measures of technique performance be free from biases introduced unintentionally by persons conducting the evaluation. The following paragraphs discuss some of the principal factors to be considered in Technique Assessment. These include:

- choice of data sets, and their preparation for analysis,
- use and treatment of Ground Truth Data, to assure compatibility with the remotely sensed imagery,
- selection of samples within the imagery to be used for training supervised classifiers to recognize particular classes, and
- methods for comparing results of different classification techniques.

5.1 ACQUISITION OF DATA SETS

The data sets to be used in evaluating classification methods should be

- sufficiently large and varied so that statistically significant numbers of data elements are present in several classes of interest,
- multivariate, since the majority of classification techniques are structured to analyze multivariate data, and
- as similar as possible to data encountered in real applications.

While data sets can be generated analytically using specified distributions, they, by nature, tend to favor some of the classification methods. Since no extensive work on the statistical modeling of disciplinary data sets has been done, it seems reasonable to test the algorithms on a few representative data sets from the particular discipline. The current work emphasizes algorithms applicable to the classification of large remotely sensed data sets such as Landsat images. Therefore, the tests here will be confined to remotely sensed image data of the earth acquired by multiband cameras and multispectral scanners. It is also necessary in evaluation to work with data sets whose ground truth is known.

The data sets are generally available as computer compatible tapes or sets of film transparencies. In the latter case, the images should be suitably digitized, and in all cases congruenced and registered before applying the digital classification algorithms.

Digitization of film transparencies requires the use of an instrument, typically a microdensitometer, that measures the optical density of each picture element (pixel) and converts it to a numerical value, generally in the range 0-63 or 0-255, the 0 value corresponding to fully transparent. These ranges are compatible with the 7- or 9-track formats employed in digital computer magnetic tape units.

To bring two or more images into congruence requires operations to be performed that assure the scene content will match in scale and orientation. Landsat multispectral scanner imagery is inherently distorted, compared with a photographic camera image, because of satellite motion and the action of the scanner. When two Landsat scenes of a given area, but acquired in different seasons, are to be compared, there is a possibility that the distortions in the later scene are slightly different from those of the earlier one. One of the scenes must then be chosen as a reference, or control, scene and the other must be "geometrically corrected" so that all its detail of shape, the outline of river banks for example, exactly matches that of the reference scene. In cases where the data have been acquired from inherently different sensors, for example the Skylab multispectral scanner and multiband camera, and must be combined for analysis, the geometric correction operations to achieve uniformity of scale and orientation are very time consuming.

Registration of two or more images requires that, once congruenced, the images may be overlaid exactly. Then the pixels that characterize an element of scene detail in the reference image can be associated one-for-one with the pixels characterizing the same element in the other images.

5.2 PREPARATION OF GROUND TRUTH INFORMATION

Ground truth information may be available in various forms. The most common form is a manually prepared map in which the various classes are marked in different colors or shadings. The information contained in the ground truth map typically is collected from different sources, for example from existing topographic maps and aerial surveys, from aerial surveys conducted specifically to acquire the ground truth information, and from in situ inspections in the field. The map should be supplemented with notes and comments, to indicate its currency and to draw attention to any factors, such as seasonal or climatic, that may affect its interpretation or validity. To achieve the ground truth detail compatible with digital computer analysis, considerable time and cost is entailed. Practical factors often preclude the collection of highly detailed ground truth information, and it is found that differences between computer classifications and ground truth arise because of the absence of this detail. The term "ground truth" therefore is applied figuratively, and the ground truth map must be used with caution in assessing the results of computer classifications.

In order to compare the classification maps with the ground truth map automatically and to highlight the differences, it is necessary to obtain the ground truth map in digital computer compatible form. Also, both the ground truth map and the classification maps should have the same scale and orientation. Since there are generally several classification maps of the same scene, it is more economical to adjust the ground truth map to match the remotely sensed data geometrically.

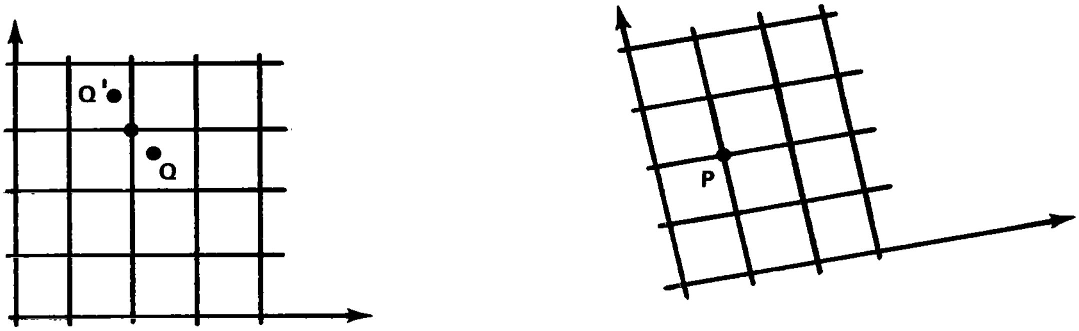
A ground truth map can be obtained in digital form by digitizing a color transparency of it on a microdensitometer, provided the map has uniform coloring for the individual classes. (A preferable method is to employ a flatbed digitizer for tracing the boundaries between classes and then to use an interior detecting algorithm to create the entire digital map.)

The geometric transformations, if needed, are found by identifying several control points on the ground truth map and the remotely sensed image. Typically, these are chosen to be landmarks with well defined edges and exhibiting high contrast against their background in the image, for example intersections (or tips) of water bodies and/or highways. Then, the transformation is required to match a set of points $A_1, A_2, \ldots, A_n$ with $B_1, B_2, \ldots, B_n$. The transformation is defined in terms of a small set of parameters which are found such that the (mean-squared) error between the two sets of points is minimized.

The transformation is then implemented digitally. This implies that the grid over which the ground truth map was originally sampled to get a digital image is to be distorted and the resulting digital map will not necessarily have its samples at integral locations on the original grid. Therefore, when the ground truth map is so/transformed, the class at a given sample location is taken to be that at the nearest sample location in the original grid. Thus, if the point P in the transformed image corresponds to the point Q in the original image, the class corresponding to P is that at Q' (see Figure 5-1).

5.3 TRAINING SITE SELECTION

As described in Section 2.0, classifiers of the supervised type require as input a set of data samples called the training set. The classes to which these samples belong must be known, and a number of samples (typically one hundred or more) must be supplied for each class which is to be identified in the data set by the classifier. These samples are used to develop decision functions, which may then be used to classify unknown samples. The classification will be reasonably



(a) ORIGINAL GRID

(b) TRANSFORMED GRID

Figure 5-1. Digital Geometric Transformation

accurate if the classes chosen are distinguishable by the measurements made in obtaining the data, the training samples are truly representative of the classes and an appropriate type of decision function is computed.

Thus, a critical aspect of the supervised classification problem is the selection of data to be used as training samples. This is generally accomplished by visual inspection of the imagery, coupled with additional sources of information such as topographic maps or personal knowledge of the area.

In order to define the segments of a digitized data set belonging to the specified classes, it is necessary to use some method of displaying the data so that the classes are recognizable. The resources required for this phase may range from a standard line printer to an interactive system employing CRT displays, with a corresponding range in ease of use and cost. In addition, it may be desirable to have the software required to display the imagery at various levels of magnification, to enhance the imagery by adjusting density levels, and to indicate on the imagery the sites from which the data are being extracted.

The initial attempt at specifying the training data sites is generally not completely satisfactory to the user. The initial classification of the data should be sufficiently accurate to indicate the outlines of the classes present much more closely than any of the input measurements. By examining the initial classification map, the user may discover that certain training sites do not lie exactly in the regions intended, due to the difficulty in discerning class boundaries in the original data. Thus, a change in the coordinates of the training sites would be desired. Also, the initial classification map may indicate areas of misclassification due to a choice of training samples for certain classes which are not representative of those classes. For example, if one wishes to define a discriminant function for the purpose of detecting forested regions, the training samples should be chosen from regions of deciduous forest and evergreen forest if both are present in the data to be classified. In this example, selection of forest training data from only evergreen data samples could well result in a loss of accuracy in deciduous forest regions, which would be evidenced in the classification results. Thus, several modifications may be made to the training samples in the course of designing a supervised classifier.

The extraction of the training data samples from the full data set is consistent with the use of a direct access storage device, as the samples will be located in small regions throughout the data set. However, a sequential access device (e.g., magnetic tape) is sufficient, even if inconvenient.

Programs employing either direct access or sequential storage have been tested on the IBM 360/65. A program which extracts up to 50 rectangular areas from a sequential data set requires (in eight-bit bytes of storage) 3.2 x 10^3 locations, plus a buffer array sufficient to store the measurements for all spectral bands along one record of data.

A set of subroutines which uses direct access storage to extract the interior of polygonal areas requires 8.4 x 10^3 bytes of storage, plus two work arrays each of length equivalent to the number of data points in a record of data, and a buffer array of length equal to the number of data points times the number of spectral bands.

5.4 ACCURACY COMPARISONS

An evaluation of the accuracy of the classification maps is a necessary part of comparing classification methods. While it is true that a ground truth map should be the basis for measurement of accuracy, it is also useful to compare classification maps with each other to find how and where they are different. A higher similarity between classification maps than relative to the ground truth map would cast doubt on the correctness or completeness of the ground truth map.

A pixel-by-pixel manual verification of the class assignments is quite a tedious exercise for all but very small scenes. Therefore, a computer algorithm is used to facilitate the determination of accuracy.

Depending upon the algorithm (supervised or unsupervised) used to generate the classification maps and on whether a comparison is being made between two classification maps or between a classification map and a ground truth map, different approaches have to be used for evaluating the similarities and displaying the differences. Three cases arise:

- known labels versus known labels (e.g., ground truth versus supervised classification),
- known labels versus unknown labels (e.g., ground truth versus unsupervised classification), and
- unknown labels versus unknown labels (e.g., two unsupervised classifications).

The details of the methods to be used in each of the above cases are described elsewhere.* The principal idea is to use "joint histograms" (contingency tables)

*"Automated Point by Point Comparison of Classification Maps," H. K. Ramapriyan, Computer Sciences Corporation Memorandum Number 5E3090-1-3, Huntsville, Alabama, July 2, 1975.

and difference maps to show the dissimilarities between maps. The joint histogram of maps 1 and 2 is defined as a matrix A with

a_{ij} = Number of simultaneous occurrences of classes i and j in maps 1 and 2, respectively.

In the case of comparisons between two maps with known labels, the "similarity measure" is defined as the total number of simultaneous occurrences of identical labels in the two maps expressed as a percentage of the total number of points in either of the maps. In terms of the joint histogram, this simply amounts to

100 Trace (A)/(Sum of all the elements of A)

Ideally, in two comparable maps, which when digitized comprise, say, 100 rows and 100 columns so containing 10,000 pixels total, the same number of pixels will be assigned in each map to a given class. If as an illustrative example, the actual land cover is 10 percent urban, 20 percent water, 40 percent agriculture, and 30 percent forest, the A matrix has values in its diagonal elements only, namely, 1000, 2000, 4000, and 3000 as in Table 5-1(a). The sum of the diagonal elements, the Trace of A, is the total number of pixels in one map, and the similarity measure clearly is 100 percent.

In practice, the two maps will not be identical in every detail and the similarity matrix shown in Table 5-1(b) is more typical. This illustrates that both maps differ from the actual land cover classification. Examining the totals, Map 1 shows 8 percent urban, 22 percent water, 45 percent agriculture, and 25 percent forest, while Map 2 shows 12 percent urban, 15 percent water, 38 percent agriculture, and 35 percent forest. Obviously, neither map is wholly accurate, and whether one is more accurate than the other can be decided only within the context of its application, since the distribution of erroneous pixel assignments, rather than the aggregate, is more significant in assessing classification accuracy. The similarity matrix is not intended to provide any measure of classification accuracy. Rather it shows explicitly the number of pixels at which the two maps agree. This number, expressed as a percentage of the total, provides the similarity measure, which in the example is 67.9 percent. In addition, by the way in which pixel differences are distributed in the off-diagonal elements, the similarity matrix highlights the ambiguities between the two classifications, and provides clues about the existence and nature of classification errors. Further insight into the distribution of errors is achieved by exhibiting the results pictorially in the form of a "difference map," examples of which are presented in Section III.

If a map with unknown labels is involved in the comparison, it is first necessary to assign labels to the class numbers in the map, and to account for the fact

Table 5-1. Illustrative A-Matrix — Joint Histogram of Maps 1 and 2

Map 2 Map 1	Urban	Water	Agriculture	Forest	Totals
Urban	1000				1000
Water		2000			2000
Agriculture			4000		4000
Forest				3000	3000
Totals	1000	2000	4000	3000	10000

(a)

Map 2 Map 1	Urban	Water	Agriculture	Forest	Totals
Urban	600	10	400	190	1200
Water	50	1200	250	0	1500
Agriculture	100	600	2890	210	3800
Forest	50	390	960	2100	3500
Totals	800	2200	4500	2500	10000

(b)

that one map may have a different number of classes present than the other. This is equivalent to performing certain elementary transformations (permuting/ adding rows and/or columns) on the matrix A. The best assignment, yielding the maximum similarity measure, is found in those cases and the resulting joint histogram and difference map are produced.

SECTION II

PROGRAM DESCRIPTIONS

1.0 INTRODUCTION

This section contains the descriptions of the programs that are used in the classification technique assessment. The programs, described herein, vary in sophistication from basic density slicing to conventional statistical methods and include the usually more complex unsupervised techniques. Hence, the density slicing method may be considered as a benchmark for obtaining some measure of cost effectiveness for the more sophisticated techniques.

The classification techniques described in this first issue are

Supervised Methods

- Density Slicing
- Maximum Likelihood
- ELLTAB
- Linear Sequential

Unsupervised Methods

- Binary Classification
- Spatial and Spectral Clustering Program (SSCP)
- HINDU

All of the mentioned programs have been made operational on the IBM 360/65 and UNIVAC 1108 at Marshall Space Flight Center, and in some cases developed, by Dr. R. Atkinson, Dr. B. Dasarathy, Mr. M. Lybanon, and Dr. H. Ramapriyan of Computer Sciences Corporation and Dr. R. Jayroe of the Information Sciences Division, Data Systems Laboratory. The authors also wish to acknowledge Mr. Clay Jones of the National Space Technology Laboratory for his cooperation and assistance in obtaining program ELLTAB.

2.0 CLASSIFICATION BY DENSITY SLICING

Density slicing refers to the process of identifying regions or objects in an image by choosing a range of densities (a density slice) corresponding to each region or object. Inspection of multiband imagery reveals that significant classes of homogeneous terrain cover can be identified visually by the reflectance characteristics and contrast within single bands. For example, bodies of water appear dark on infrared bands while some types of pasture appear very bright. Similarly forest canopy appears dark in the red band, while man-made objects, for example, concrete or asphalt paving, buildings and large stone or metal structures, appear bright in the green band. Accordingly, terrain cover types can frequently be separated on the basis of the contrast against their background in any one spectral band. Human photointerpretation depends heavily on such tonal differences for discriminating between different object classes in a scene. The process is frequently combined with color coding, in which the density slices are made highly visible by being assigned different colors. The method is appealing because of its simplicity, since cross-correlation of the reflectance values between several spectral bands is not required. Thus the classification process does not involve numerical computations employing discriminant functions followed by a decision based on the result, but may be implemented by testing the density values against the density ranges for each class. However, a density slicing result may be obtained by using a linear classifier restricted to the use of one spectral band.

Correspondingly, the density ranges can be chosen manually by examining the density values in each region of interest, or the spectral band and density range for each class may be selected by a feature selection and linear classification algorithm. The latter method was tested for inclusion in this report as the algorithms were available. It is worth noting that this procedure does not result in the most efficient implementation of the density slicing technique. Indeed special equipment is commercially available, in which the density slicing function is performed at very high speed by electronic comparator circuits, and the density slices displayed in color code on a television type display screen.

2.1 RESOURCE REQUIREMENTS

The classification system requires as input the number of classes and spectral bands present in the data and a set of training samples.

The data samples to be classified are to be supplied with the measurements for each spectral band arranged in vector format, and the classification results are written on an output tape.

The following arrays are required in main storage:

- An array to store the training samples indexed by class number, feature number, and sample number

- An array into which all the data in one scan line can be read, before classification

- An output array containing the class numbers for each set of measurements in a scan line

- Arrays to store the discriminant coefficients by class, and the order in which discriminants are tested

- A work array for storing the interclass and intraclass distances for each feature

- Two work arrays dimensioned by the number of training samples, for use in training

The program subroutines required to perform various tasks are given in Table 2-1. Their purpose and storage requirements are included. These requirements are constant, as storage for the problem-dependent arrays listed previously is allocated in the main control program.

2.2 ANALYSIS PROCESS

In order to verify that the best possible spectral band is chosen to discriminate any given class from all the others, a quantitative band (feature) selection method is applied first. Usually this confirms what is visually obvious, but sometimes the quantitative selection scheme will identify a spectral band to be superior for class discrimination than another that appears from visual inspection to be appropriate.

Using a set of training data samples whose classifications are known, the average distances between pairs of samples from different classes and within classes were computed for each spectral band. The spectral band chosen is that for which the ratio of between-class distance to within-class distance is a maximum.

Linear discriminant functions are then computed for each class, using the spectral band chosen by the above criterion for each class. The coefficients in the discriminant function are chosen by an iterative procedure.* The two coefficients determined (constant term and data value multiplier) may be used in a linear discriminant function, as is done in this case, or may be used to calculate the density ranges occupied by each class.

*An Integrated Feature Selection and Supervised Learning Scheme for Fast Computer Classification of Multi-Spectral Data," A. D. Bond and R. J. Atkinson, Remote Sensing of Earth Resources, Vol. 1, F. Shahrokhi, Ed., U. of Tenn. Space Institute, March 1972.

Table 2-1. Density Slicing Classifier

Subroutine	Purpose	360/65 Storage (8-bit bytes)
EFFECT	Compute interclass and intraclass distances, determine order of class separability.	3.8×10^3
SNOPAL	Supervised nonparametric learning of linear discriminant functions.	4.8×10^3
NTEST	Test classification of training samples.	1.5×10^3
NCLASS	Handle I/O to classify data; compute class percentage occupancies.	1.4×10^3
NOPACA	Linear discriminant classification of a feature vector.	7.5×10^2
GASINV	Matrix inversion; required in learning algorithm.	1.8×10^3
SORTSL	Sorting algorithm, required in class ordering	1.1×10^3

2.3 PERFORMANCE CHARACTERISTICS

The algorithm can operate on large numbers of spectral bands and classes. The size of the data set to be classified is immaterial, as the classification is done on a point-by-point basis.

The output of the program is a tape containing the class number of each pixel and a listing of the class populations and percentages.

A data set containing 52,000 pixels was classified in 16-1/2 seconds on the IBM 360/65.

3.0 MAXIMUM LIKELIHOOD CLASSIFIER

The maximum likelihood classifier is a supervised, parametric technique and is probably the most widely known and used multichannel data classification method.* A set of data samples, whose classifications are known, is required to define the parameters of the functions which are used to determine the classes of unknown data samples. The required parameters are those which define the Gaussian distributions for each class of the training data, namely the mean vectors and covariance matrices.

3.1 RESOURCE REQUIREMENTS

The maximum likelihood classification system requires as input parameters the number of classes and features (spectral bands) present in the data and the Gaussian parameters (mean vectors and covariance matrices). The data samples to be classified are assumed to reside in feature vector arrangement (typically the reflectance measurements pertaining to any one pixel organized in the order of spectral bands or channels, green, red, infrared, etc., followed by the measurement pertaining to the adjacent pixel) on a FORTRAN readable data set. The classification results, in the form of numbers (1-n) corresponding to n classes present, are written on an output tape.

The following arrays require an amount of storage dependent on the number of classes and spectral bands in the data, the number of training samples selected (typically 100), and the number of pixels in a line of data on the input and output data sets:

- an array to store the training samples by class number, feature number, and sample number,
- an array into which the input feature vectors are read,
- an array in which the output class numbers are placed, to be written as output, and
- arrays to store the mean vectors and covariance matrices by class number.

The program subroutines required to perform the various tasks, such as computing the mean vectors and covariance matrices, handling input/output of data and

*"Learning Machines," N. J. Nilson, McGraw-Hill, N.Y., 1965.
"Information Processing of Remotely Sensed Agricultural Data," Proceedings IEEE, Vol. 57, No. 4, April 1969.

class numbers, and classification, are given in the following table, along with their storage requirements. These requirements are constant, as storage for the problem-dependent arrays listed previously is allocated in the main control program.

Table 3-1. Maximum Likelihood Classifier

Subroutine	Purpose	IBM 360/65 Storage (8-bit bytes)
SUBLOP	Compute Gaussian parameters from training samples	2.0×10^3
PTEST	Test classification of training samples	1.4×10^3
.PCLASS	Handle I/O to classify unknown data set; compute class occupancies	1.3×10^3
MALICA	Maximum likelihood classification of a feature vector	9.3×10^2
GASINV	Matrix inversion; inverts covariance matrix	1.8×10^3

3.2 ANALYSIS PROCESS

The following paragraphs describe the form of the Gaussian distribution, the definition of parameters, and the classifier.

If a set of data samples from a single spectral band is examined, the number of measurements falling in successive small intervals may be represented by the height of the bars on a histogram, as illustrated in Figure 3-1.

A smooth curve outlining the shape of the histogram is a probability distribution curve. A typical example is the bell-shaped curve of the Gaussian or normal distribution, as in Figure 3-2.

The mathematical function for this curve is

$$P(x_j/c_i) = \frac{1}{\sqrt{2\pi}\,\sigma_j} \exp\left[-\frac{1}{2}\left(\frac{x_j-m_j}{\sigma_j}\right)^2\right]$$

where σ_j and m_j are the standard deviation and mean for measurement x_j belonging to class c_i.

Considering multichannel measurements, the joint probability function for a complete multivariate feature vector is

$$P(x_1,x_2,x_3,\ldots,x_n/c_i) = \frac{1}{\sqrt{(2\pi)^n D}} \exp\left[-\frac{1}{2}(X-M)^T K^{-1} (X-M)\right]$$

where (X-M) is the vector $\{x_1-m_1, x_2-m_2, \ldots, x_n-m_n\}$, K is the covariance matrix, and D is the determinant of K. The elements of the covariance matrix are a measure of the deviation of the corresponding x's from their mean values m:

$$K_{ij} = \frac{1}{N-1} \sum_{n=1}^{N} (x_{in} - m_i)(x_{jn} - m_j)$$

where N is the number of data samples used in the calculation.

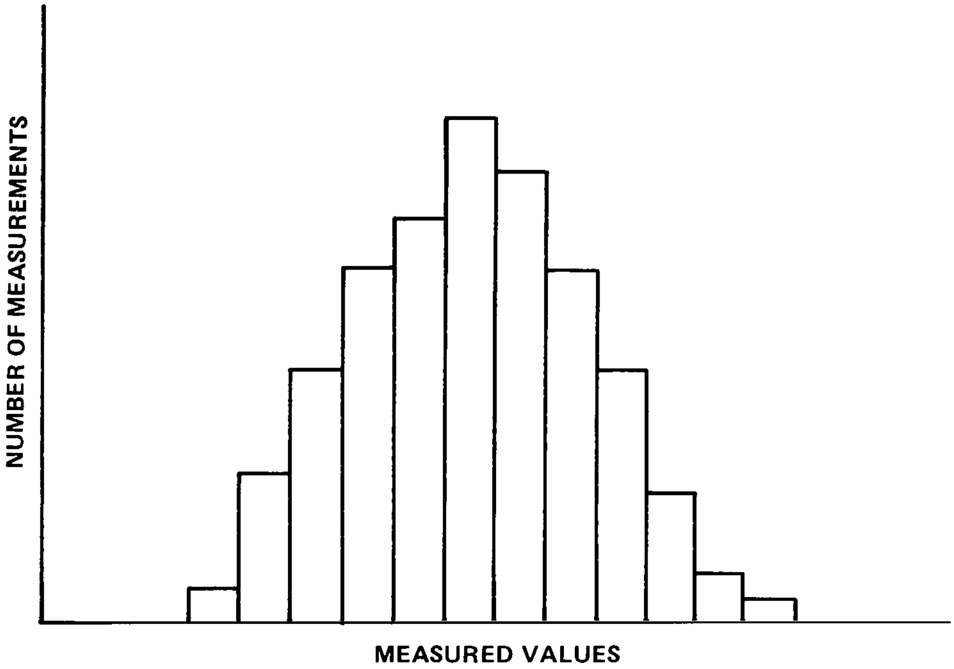


Figure 3-1. Histogram of a Set of Measurements

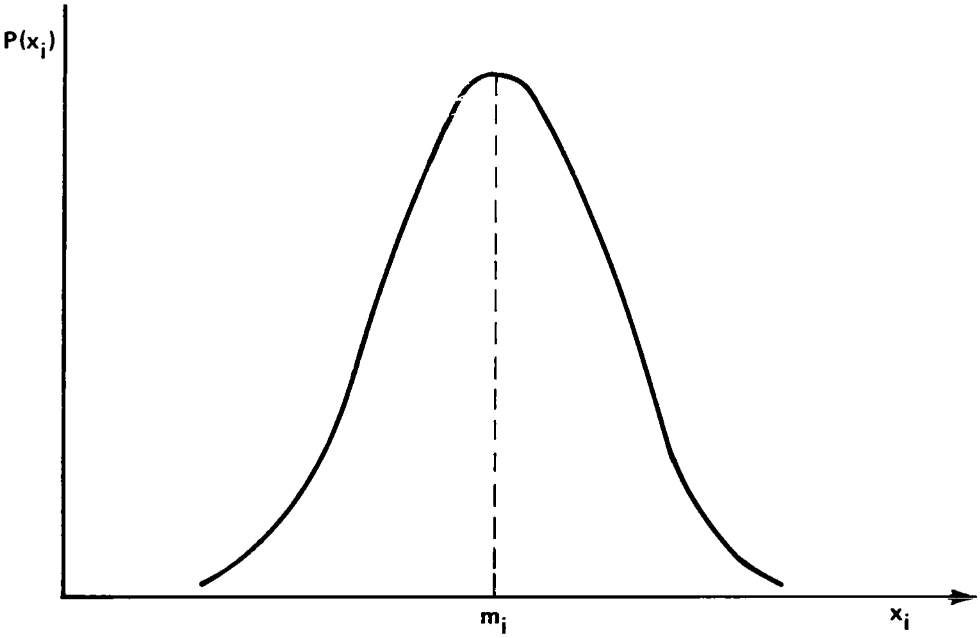


Figure 3-2. Gaussian Probability Distribution

The parameters--mean values and covariance matrices--completely define the Gaussian distribution functions. These parameters are easily determined for each class under consideration from the known set of training samples.

When the Gaussian parameters have been estimated, the Gaussian probability distribution for each class is completely defined. Thus, given any unknown feature vector, it is possible to compute the probability of this feature vector belonging to any one of the classes under consideration. Assignment is made to the class for which the probability is greatest; this is termed the maximum likelihood method of classification. For faster computation, the logarithm of the probability is computed and the decision function takes the form

$$G_i = \ell n\, P_1 - \frac{1}{2} \ell n\, |K_i| - \frac{1}{2} (X-M_i)^T K_i^{-1} (X-M_i)$$

P_i is the probability of class i being present, M_i is the mean vector, and K_i is the covariance matrix. The decision point between two classes occurs when the probabilities are equal, at point x_0 in Figure 3-3. Note that x_0 is not midway between the means when the widths of the distributions are unequal. At point x_1, the probability P_1 is greater and hence x_1 is assigned to class 1.

The analysis flow in classifier training is shown in Figure 3-4. Once a candidate set of training samples has been identified, the Gaussian parameters (mean vector and covariance matrix) are computed. Then using the above decision function, or discriminant, each training sample is classified as if its true identity were unknown. The test results are printed and scrutinized. Ideally, all the training samples that were selected from a particular class will be assigned by the classifier to that class. If this is not the case, the reason for the variation must be determined. Often the variation will be due to impurity in the training samples, caused by inadvertent selection of samples lying on a class boundary. In this case, the training sample set must be refined by choice of other more representative samples. A further cause of variation is that often the distribution of training samples is not well approximated by a Gaussian distribution. If the histogram of the training samples for one class shows several peaks, the data is said to be multimodal, and in this case, the class should be divided into subclasses, each one of which is characterized by a set of Gaussian parameters.

When the analyst is satisfied that the training sample set is the best attainable, the entire image data set is classified, as in Figure 3-5.

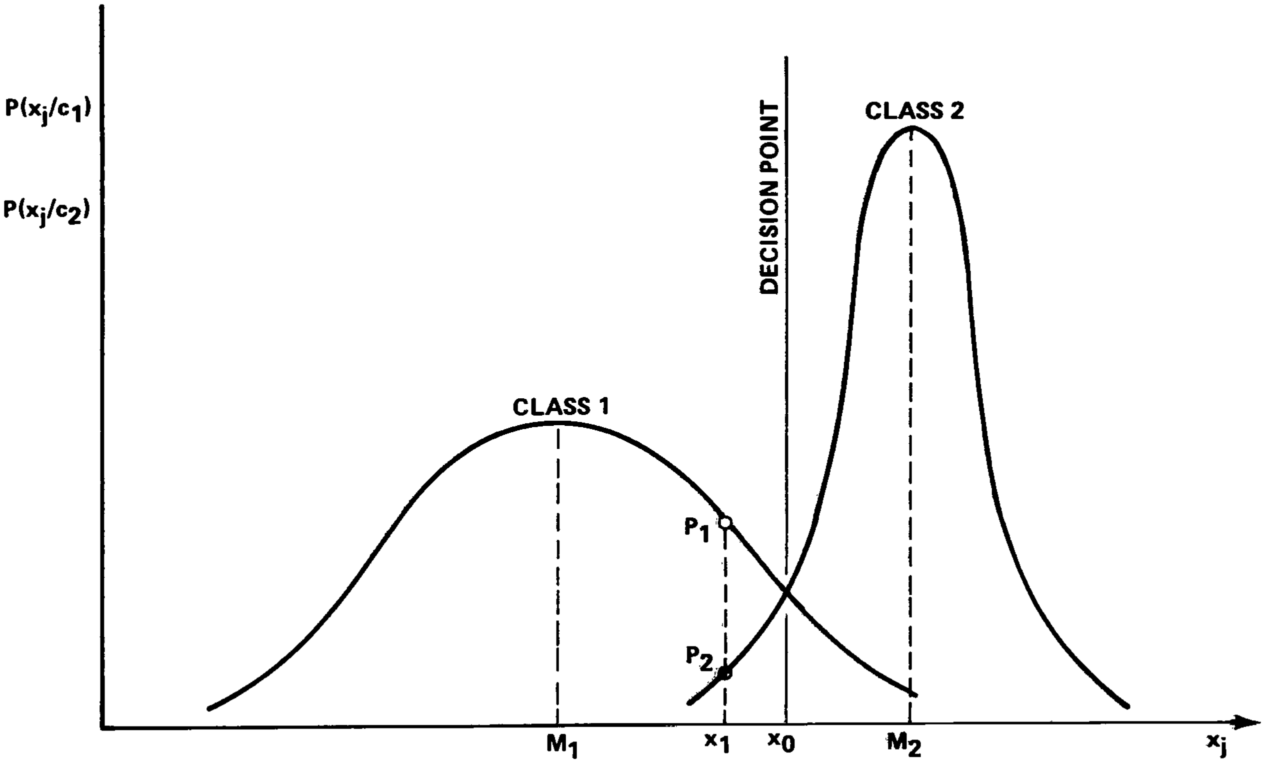


Figure 3-3. Decision Point for Assigning Measurements to Two Gaussian Distributions

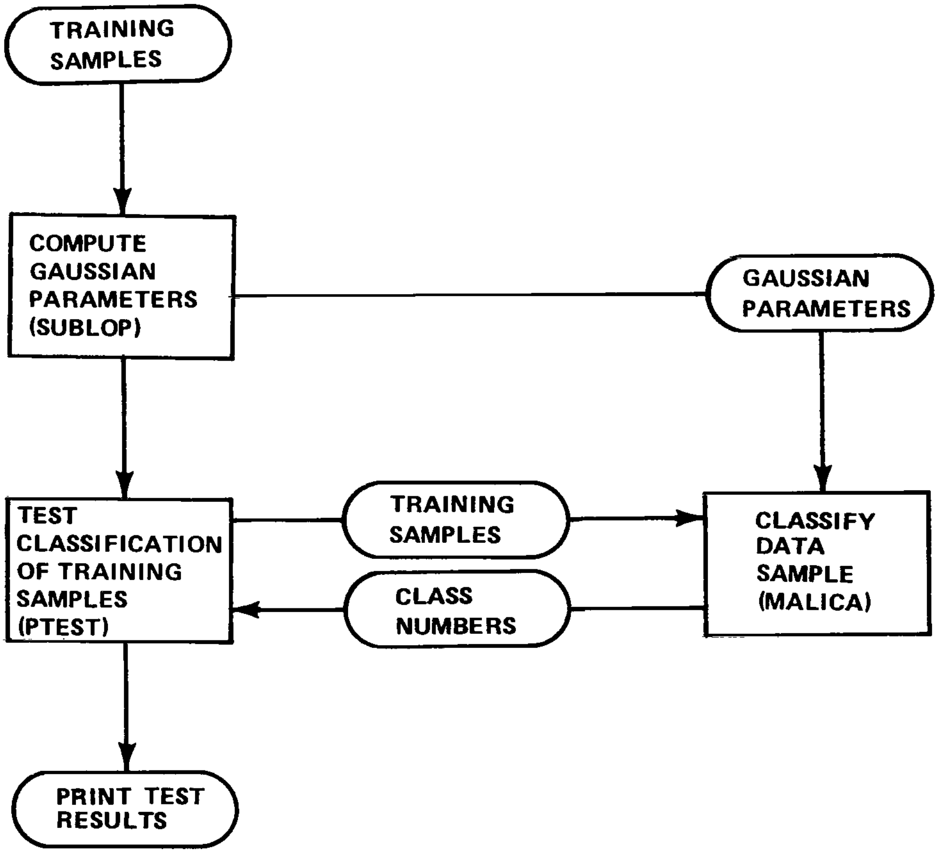

TRAINING SAMPLES
COMPUTE GAUSSIAN PARAMETERS (SUBLOP)
GAUSSIAN PARAMETERS
TEST CLASSIFICATION OF TRAINING SAMPLES (PTEST)
TRAINING SAMPLES
CLASS NUMBERS
CLASSIFY DATA SAMPLE (MALICA)
PRINT TEST RESULTS

Figure 3-4. Classifier Training Phase of Maximum Likelihood Classification

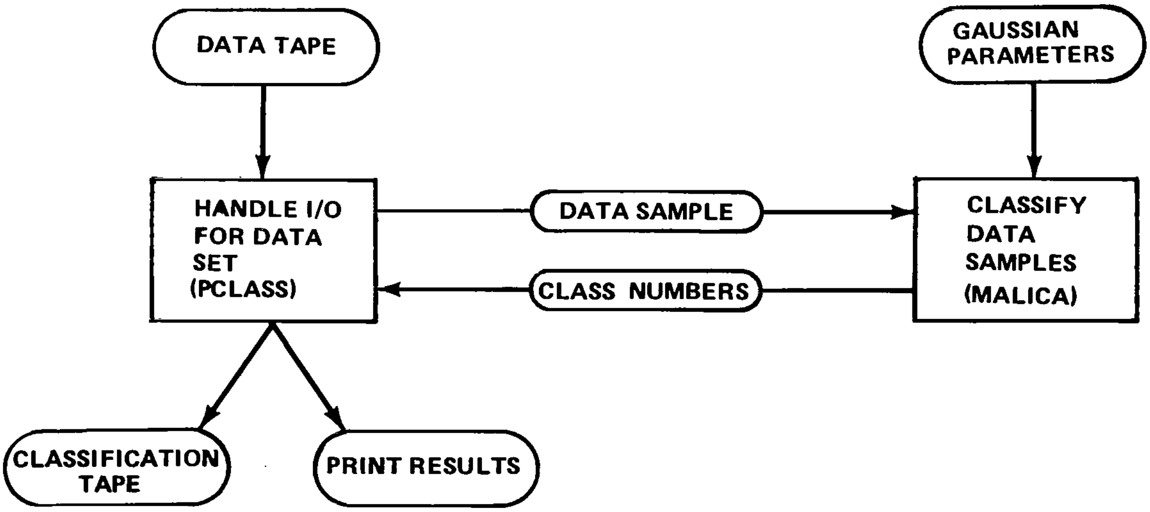


Figure 3-5. Classification Phase of Maximum Likelihood Classifier

3.3 PERFORMANCE CHARACTERISTICS

The performance of a maximum likelihood classifier with respect to accuracy and speed may be inferred from an examination of the method itself. If the data samples do obey the Gaussian distribution for each class, this method produces optimum results. However, the actual data samples belonging to a given class may produce a histogram having two or more peaks. Typical causes of this effect in earth resources data are differing soil conditions, sun angle, crop health and maturity, and the widely varying reflectivity of man-made objects. In the case of such a multimodal distribution, the Gaussian parameters which are computed do not accurately describe the actual distribution, and the classification accuracy is reduced.

The maximum likelihood classifier is relatively slow because the classification of a data sample requires the evaluation of the decision function for each class being considered.

This method will operate satisfactorily on large numbers of spectral bands and classes. The size of the data set to be classified is immaterial to the process, as each data point is classified independently.

The output of the classification program is a tape containing the classification map and a listing of the class populations and percentages.

The following table gives classification rates for various data sets. Computer time required to read the input tape and write the classification tape is not included.

Table 3-2. Maximum Likelihood Classifier Times (IBM 360/65)

Number of Classes	Number of Spectral Bands	Number of Data Points	Computer Time (Seconds)	Points Classified Per Second
3	4	510,000	564	904
6	4	52,000	114	457
7	3	122,850	225	546
6	16	52,000	690	75
7	16	52,000	809	64

4.0 ELLTAB

The name ELLTAB stands for ELLiptical TABle, which gives a partial description of the program. ELLTAB is a version of the (supervised) Gaussian maximum likelihood method, implemented using a novel table lookup technique. The program is an application of the general table lookup pattern recognition method devised by Eppler.* The general idea of the method is, in the training phase, to precompute the possible results of the decision rule, as a function of position in feature (measurement) space, and store them in a table. Then, in the classification phase, each measurement vector is used to enter the table, which tells which class the point is to be assigned to. (The method is described in more detail in Section 4.2.)

In constructing the table, each possible result only needs to be computed once, while in conventional implementations of pattern recognition techniques, the same calculation could be performed several times. For four-dimensional data for which each component may take on all integer values from 0 to 255, such as LANDSAT multispectral scanner data, there are about 4.29 x 10^9 conceivable data values. If it were necessary to calculate results for each of these values, the table lookup method would probably be slower than any other method. However, by making use of the statistics of the training data set, ELLTAB considerably reduces the amount of computation. In fact, in most cases it is not even necessary to calculate the probabilities of assignment of points to the various classes. The time per point required for the classification phase is approximately proportional to the number of classes. (Actually the increase in time as the number of classes is increased is slower than direct proportion.) Since the classification itself is performed simply by looking up results in a table, the time required is not at all dependent on the classification rule used in preparing the table. So, for very large data sets, particularly if the classification rule requires complex calculations in a conventional implementation, and for many classes, the table lookup method could be expected to provide an important increase in speed.

4.1 RESOURCES REQUIRED TO RUN ELLTAB

ELLTAB was originally written in FORTRAN V for the UNIVAC 1108 computer. It is presently being tested here in its 1108 version. While it is certainly true that testing of several routines on the same computer provides a basis for comparison, it must be kept in mind that it is not an absolute comparison. If program A is better in some sense than program B on one computer, the reverse

*"Table Look-Up Approach to Pattern Recognition," W. G. Eppler et al, Proc. 7th International Symposium on Remote Sensing of Environment, U. of Michigan, Ann Arbor, May 1971.

may be true on another computer. Even though a program is written in a high-level language, such as FORTRAN, some machine-dependent features (a greater or lesser number of registers, special instructions, etc.) may affect performance. And special care is required to avoid the use of convenient machine- or system-dependent features available on the computer for which the program is first written. Because of such factors, conversion of a program from one computer to another may be much less than straightforward, and unless considerable time and effort are expended an inferior version of the program might result. For these reasons, ELLTAB was first tested on the 1108.

The version of ELLTAB received from the author* uses several 1108 features that are not available on the IBM 360/65 or other computers.** These include:

- The FLD (bit-manipulation function)
- Use of the FLD function on the left of the equal sign in arithmetic statements
- The BOOL (make typeless) function
- Backward DO loops (negative increment)
- DO loops starting with zero index
- RETURN 0 statement
- DEFINE statement
- PARAMETER variables
- ERTRAN (system processor)
- NTRAN (system processor)
- Use of literals on the right of the equal sign in arithmetic statements
- O (octal) format (the nearest equivalent on the 360 is the Z format)

*"Implementation of an Advanced Table Look-Up Classifier for Large Area Land-Use Classification," Clay Jones, Proc. 9th International Symposium on Remote Sensing of Environment, U. of Michigan, Ann Arbor, 1974.

**UNIVAC 1100 Series, FORTRAN V Programmer Reference, UP-4060 Rev. 2, Sperry Rand Corporation, 1973.

Certain of these could be simulated on the 360, but others would require extensive reprogramming. A good example of the latter is the use of the FLD function. That function makes possible the movement of specific bit strings in a FORTRAN program. Although such a program could be written for another computer (most conveniently in assembly language), the appearance of FLD on the left of the equal sign in arithmetic statements would be considered a syntax error by the FORTRAN compiler. Also, the uses to which the FLD function are put in ELLTAB are specialized. They include unpacking nine 8-bit bytes from two 36-bit words, storing and retrieving lookup table information (values are stored in partial words to save space), and packing output values to prepare a tape for a specific output device. On a byte-oriented computer (with 32-bit rather than 36-bit words) such as the 360, most of this would be irrelevant or would be done entirely differently. Also, one part of ELLTAB deals with converting EBCDIC annotation data to FIELDATA--unnecessary on the 360. In short, ELLTAB is explicitly an 1108 program, despite being entirely in FORTRAN. A potential user would either have to run it on the 1108 or invest considerable effort in conversion and reprogramming.

ELLTAB consists of two executable modules, ELIPSE and ASSIGN. Each contains a main program and several subroutines. ELIPSE constructs the lookup table (training phase), which is then used by ASSIGN to classify a scene (table lookup phase). The table constructed by ELIPSE is based on partitioning feature space into hyperellipsoids, one for each class, based on statistics derived from training data; the Gaussian maximum likelihood classification rule is implemented. The program allows for the possibility of overlap of ellipsoids, which sometimes results in multiple ranges of a feature for a single class. ASSIGN makes use of the table "built" by ELIPSE in classifying data, and outputs a classification tape. Because ASSIGN merely reads a table in a standard format, it should be able to accommodate tables constructed using other classification rules.

The two modules are executed separately. ELIPSE requires about 30K words of core storage. About 70 percent of this space is used for data storage, and over half of the space occupied by data is accounted for by three arrays: one to hold the table built for a single class, another to hold the table after an operation known as "null squeeze," and one to hold the inverse covariance matrices for all classes other than the one for which the current table is being constructed. An additional 2K words are used to hold covariance matrices, in the original order and sorted as needed in building the table (in general, the order differs from table to table). The dimensions of these arrays depend on the maximum size expected for a table, the maximum number of classes the program can accommodate, and the number of channels. One tape drive is required for the (output) table tape. The tape is written using NTRAN, an efficient I/O system permitting parallel processing. All other output is on the system output device (usually a line printer). Input to ELIPSE is on cards. The information required is a set of "switches" (0 or 1 values) to direct the program with regard to six output options, the number of classes to be used, the number of scanner channels (restricted to 4 in the present implementation), and the minimum and maximum scanner output values. In addition, some information is separately required for each class: an identifier, class number, training statistics (mean vector and covariance matrix), a priori probability of assignment (default value is 1/number of classes), and a "relax option" and "quadratic threshold," which are discussed in Section 4.2. An optional input quantity is the number of points in the training data set, printed for convenience along with the other input data. No direct-access storage or special output devices are used.

It should be noted that the module ELIPSE does not start with training data. Rather, it requires as input training statistics (means and covariances), which must be derived from training data separately. Therefore, ELLTAB is not a complete system. Perhaps this can be regarded as avoiding duplication, since a user is likely to already have a program for deriving statistics from training data. However, in order to use ELLTAB, a user must have such a program.

The other executable module, ASSIGN, requires about 27K words of core storage (modified for local use as described below). The program is this size with the array used to hold the combined lookup table for all classes dimensioned 9000; as received, it was dimensioned 6500. Data storage was over 70 percent of the total, and nearly two-thirds of that was taken up by two arrays, the combined lookup table and an array to hold a line of unpacked (vector) data. Three other arrays (one record read from table tape, one line of packed data, and one classified line) bring the total for the five arrays to 75 percent of the space used in ASSIGN for data storage. Three tapes are used: the table tape, the (input) data tape, and the (output) classification tape. However, only two tapes are required simultaneously. After the table tape is read into core, it is dynamically freed and the data tape is dynamically assigned, using ERTRAN (an 1108 EXEC 8 system feature that allows "executive requests"--requests to use a feature of the operating system--to be made directly from the user program). No direct-access storage is used by ASSIGN. The special system routines NTRAN and ERTRAN are called by ASSIGN. As received, ASSIGN wrote an output tape in a special format for a specific output display device. For the purposes of this study, that routine was modified to simply produce a tape containing the point-by-point class assignment numbers, one scan line per record. This is actually more general since the interface with a specific output device can be implemented fairly trivially as a post-processor (or as several post-processors: one for a filmwriter, one for printer plots, one as input to a data base, etc.). Printed output consists of:

- a summary listing of the card input data,

- information decoded from the Landsat bulk data tape's ID record,

- physical description of output tape format (produced by the new tape output subroutine), and
- class assignment histogram (number of pixels assigned to each class).

The input data on cards consists of:

- extent of the data to be processed (first and last records, first and last pixels),
- total number of classes for which tables have been constructed, and
- number and identification of classes into which points are to be classified (may be less than total).

Some other information specific to the program is also input.

4.2 ANALYSIS PROCESS

The table lookup method of pattern recognition is motivated by a desire to reduce the total amount of computation required for classifying large data sets, possibly using complex decision rules. A secondary goal is to make it possible to accomplish this using minicomputers. After a step that partitions feature space into regions according to some decision rule and constructs tables incorporating this information, classification of multispectral data is performed simply by entering the tables, which have a form essentially independent of the decision rule.

ELLTAB is an implementation of the table lookup technique for a specific classification method. The method is the supervised Gaussian maximum likelihood technique, with "quadratic thresholds" (defined below). The description of the algorithm will begin with a sketch of how the table lookup is performed, and then outline the method of constructing the tables. So first the operation of ASSIGN will be described, and then the operation of ELIPSE. It will be seen that the method of the former is more general than the latter. Also, the methods are somewhat more general than the programs themselves. Figure 4-1 illustrates the flow of data through ELLTAB, while Figure 4-2 shows the lookup procedure.

The tables constructed and used by ELLTAB comprise a geometric description of the classes in feature space. The lookup portion of the program (ASSIGN) is independent of the method used to partition the measurement space among the various classes (ELIPSE). There is one table for each class. Processing of each data point begins by forming a hypothesis C concerning the class assignment. The initial hypothesis is that the class is the same as that assigned to

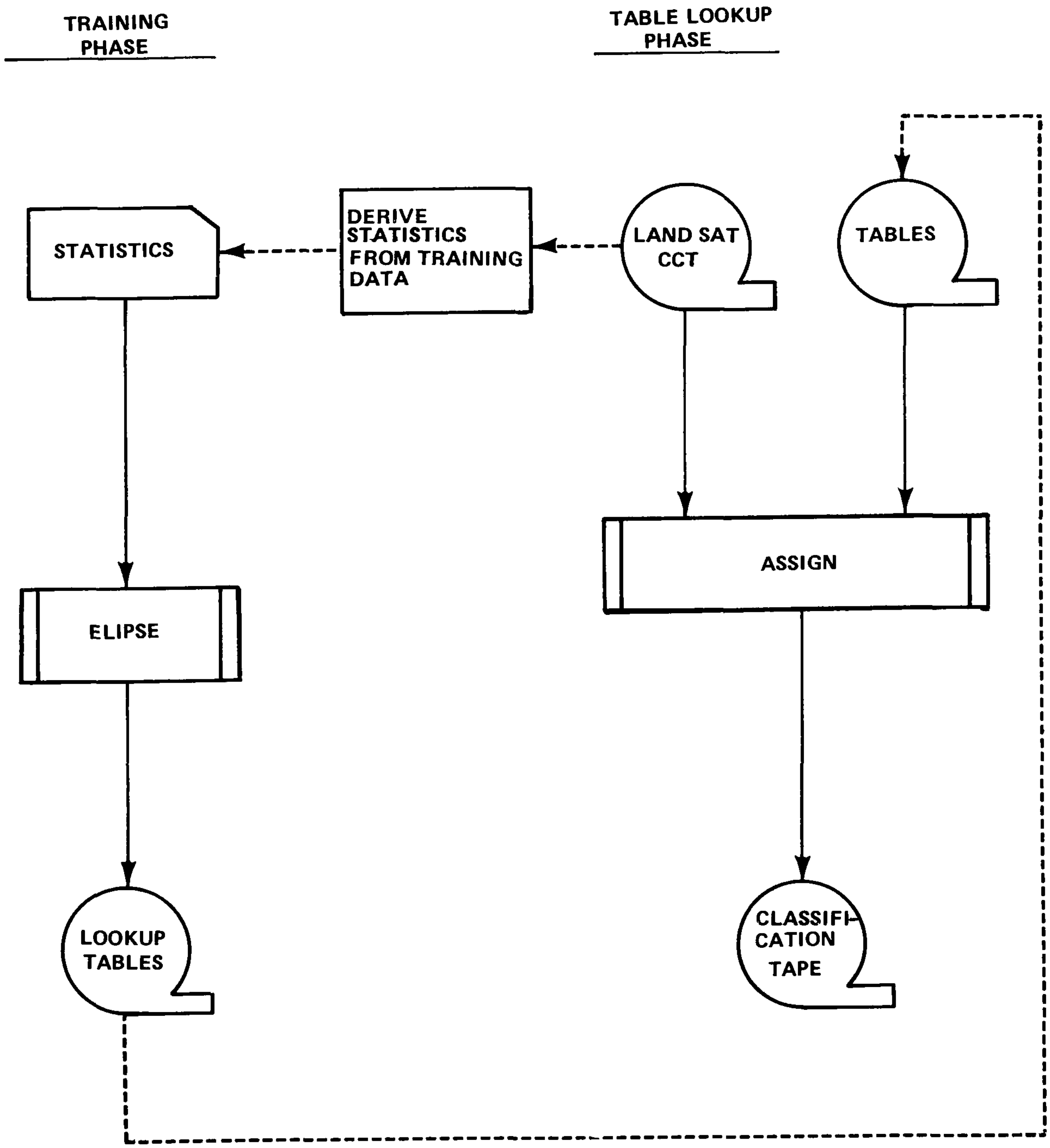
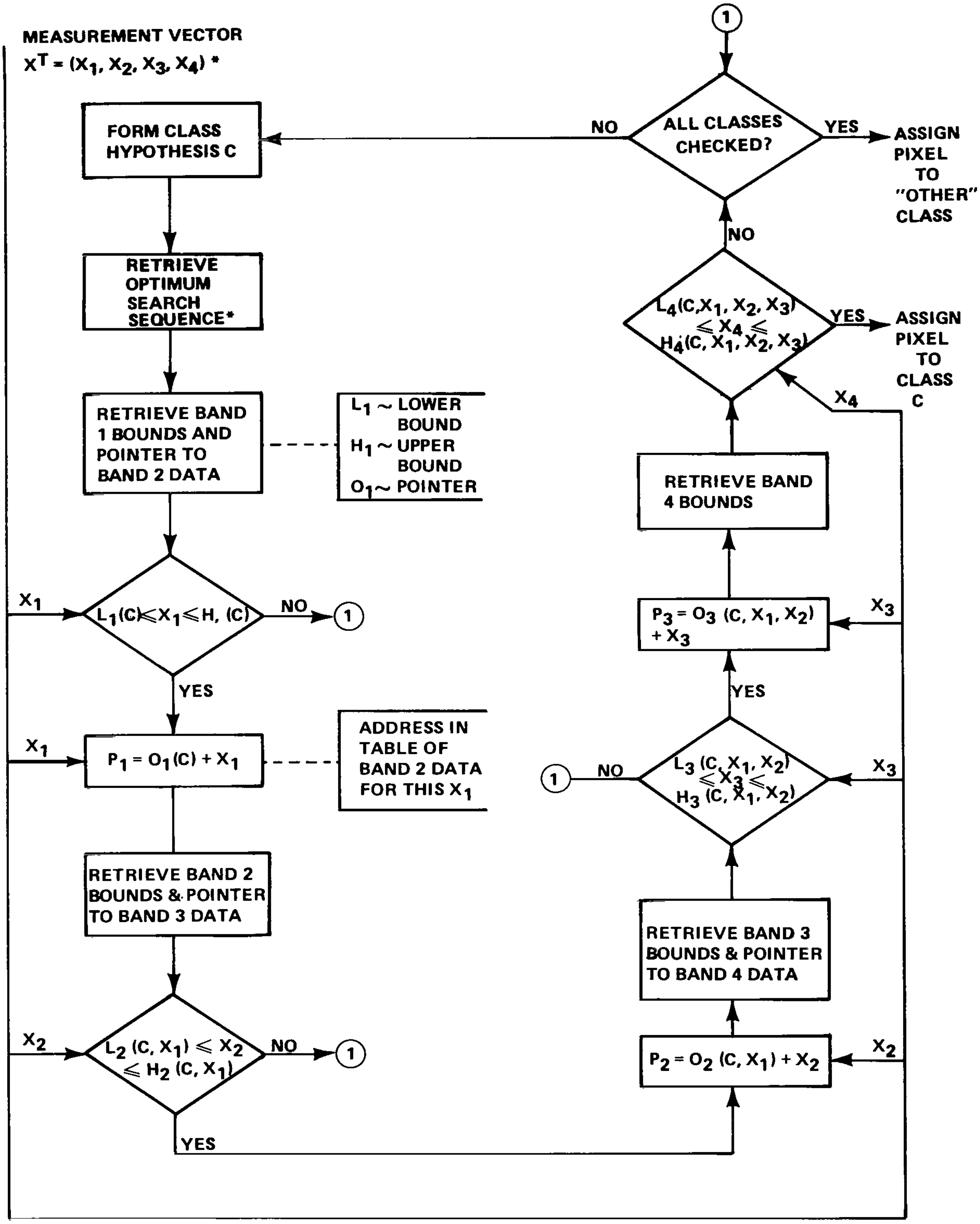


Figure 4-1. Flow of Data Through Modules of ELLTAB


MEASUREMENT VECTOR
$X^T = (X_1, X_2, X_3, X_4)$ *
FORM CLASS HYPOTHESIS C
RETRIEVE OPTIMUM SEARCH SEQUENCE*
RETRIEVE BAND 1 BOUNDS AND POINTER TO BAND 2 DATA
L_1 ~ LOWER BOUND
H_1 ~ UPPER BOUND
O_1 ~ POINTER
X_1
$L_1(C) \leq X_1 \leq H_1(C)$
NO
1
YES
X_1
$P_1 = O_1(C) + X_1$
ADDRESS IN TABLE OF BAND 2 DATA FOR THIS X_1
RETRIEVE BAND 2 BOUNDS & POINTER TO BAND 3 DATA
X_2
$L_2(C, X_1) \leq X_2 \leq H_2(C, X_1)$
NO
1
YES
1
NO
ALL CLASSES CHECKED?
YES
ASSIGN PIXEL TO "OTHER" CLASS
NO
$L_4(C, X_1, X_2, X_3) \leq X_4 \leq H_4(C, X_1, X_2, X_3)$
YES
ASSIGN PIXEL TO CLASS C
X_4
RETRIEVE BAND 4 BOUNDS
$P_3 = O_3(C, X_1, X_2) + X_3$
X_3
YES
1
NO
$L_3(C, X_1, X_2) \leq X_3 \leq H_3(C, X_1, X_2)$
X_3
RETRIEVE BAND 3 BOUNDS & POINTER TO BAND 4 DATA
$P_2 = O_2(C, X_1) + X_2$
X_2

*COMPONENTS MAY BE ORDERED DIFFERENTLY FOR EACH CLASS C

Figure 4-2. Table Look-up Procedure Used in ELLTAB

the preceding pixel. If that hypothesis fails, the other classes are tested in order of decreasing a priori probability. (It is possible that a point will not be assigned to any class.) The testing is done as follows: The first component, X_1, of the point is tested to see whether it lies within the permissible range of values for that class, $L_1(C) \leq X_1 \leq H_1(C)$. If not, that hypothesis fails. If so, X_2 is tested to see whether it lies in the allowed range for that class and for that value X_1, $L_2(C,X_1) \leq X_2 \leq H_2(C,X_1)$. If that test is also passed, X_3 is tested to see if it lies in the range $L_3(C,X_1,X_2) \leq X_3 \leq H_3(C,X_1,X_2)$; etc. The tables, then, contain a description of the class boundaries, along with "pointers" to tell where in the tables to look to find the limits for the next test. In case some of the classes overlap, it is possible that, for a given class C, the allowable range for a component may be discontinuous--that is, there may be a gap in it. ELLTAB allows for this eventuality. Also, for different classes the labels X_1, X_2, ... may label different components. The order of utilizing components of measurements is chosen for each class to minimize the size of the table for that class. Eppler* proposed using the table lookup method in four dimensions; the amount of space required to store the tables increases dramatically for higher dimensionality. He also proposed using a feature selection technique, for $N > 4$ dimensions, to select the best (possibly different) set of four channels to analyze for each class. ASSIGN (and ELIPSE) is a four-dimensional program. However, it does not incorporate this latter suggestion.

The table-building phase could use any method of partitioning measurement space and constructing tables in the format described above. ASSIGN explicitly uses the Gaussian maximum likelihood method. The tables describe hyperellipsoids in four-dimensional space. Assuming first that the regions for the classes do not overlap, the statistics derived from training data are used to determine the ellipsoids. The sizes are given by the quadratic threshold values Q specified in the input data. The parameter Q is the maximum Mahalanobis distance from the mean that a measurement vector can have and still be assigned to that class. Values of Q are equal to percentage points of the χ^2 distribution--that is, values of χ^2 (for four degrees of freedom) for specified exclusion probabilities. A value $Q = 13.2767$ will exclude 1 percent of the sample points from a true normal distribution. Table size is sensitive to the value of Q. If there is no overlap between classes, nothing else is necessary. For regions of overlap, points are assigned to the class for which the likelihood discriminant function

*"An Improved Version of the Table Look-Up Algorithm for Pattern Recognition," W. G. Eppler, Proc. 9th International Symposium on Remote Sensing of Environment, U. of Michigan, Ann Arbor, April 1974.

has the greatest value. It is possible for an anomalous condition to occur for certain cases of overlap. An ellipsoid for a class A may be partially or completely contained within the region nominally belonging to a larger ellipsoid for another class B. It is possible that a point outside the pre-specified quadratic threshold Q for class A may nonetheless have a greater likelihood of belonging to A than to B. Therefore, it will not be assigned to either class. The relax option permits the program to relax the strict assignment rule and assign the point to class B, in such cases.

4.3 PERFORMANCE CHARACTERISTICS

Since ELLTAB is an implementation of the multivariate Gaussian maximum likelihood decision rule, its performance (e.g., with regard to the type of classification errors it may yield, etc.) should be similar to that of other implementations of the method. Because of the quadratic threshold feature described in Section 4.2, some data points will generally be assigned to the unclassified "class." That is, while in programs without this feature every pixel will be assigned to some class, certain points will not be assigned to any class by ELLTAB.

The following statistics refer to ELLTAB as it was received from the author, except where otherwise noted:

Number of channels = 4 (fixed)

Maximum number of classes = 100

Maximum size of a table for a single class = 5000 words (in ELIPSE)

Maximum size of combined table for all classes = 9000 words* (in ASSIGN)

The last two quantities, although not directly comparable with anything in other programs, are listed because they have a major effect on the amount of storage required for ELLTAB. In connection with this, Eppler asserts that the table lookup method is probably not practical for more than four dimensions because of the amount of storage that would be necessary for the tables.

*This dimension was originally 6500, but it needed to be increased to run the test cases.

There is essentially no limit to the number of data points ELLTAB can process in a single run, since it classifies one scan line at a time. ELLTAB was specifically written to process Landsat data; it "expects" its data tape to be in Landsat bulk data tape format. The array that holds one packed scan line (nine 8-bit bytes in each two consecutive 36-bit words) is dimensioned 733 in the subroutine in which it is defined. This is sufficient for standard Landsat tapes (810 four-dimensional data values, followed by 56 calibration values, packed 4-1/2 per computer word). However, the subroutine that is used to read records from the tape permits 752 words to be read. If the number of words in a record exceeds 733, this could lead to unpredictable errors or failure. (The tape read subroutine also has other defects.)

In addition, the array to hold an unpacked record and the array to hold a line of classified output are dimensioned to hold 876 points. This number is not consistent with either of the dimensions above. Since the size of a packed record is actually restricted to 733 (containing 810 pixels), the 876-point arrays waste storage space.

The documentation provided with ELLTAB was not adequate to permit a user unfamiliar with the details of the program to use it without a period of experimentation. The documentation consisted of one-sentence descriptions of each routine, reproductions of program listings (with some pages missing and some out of order), sample input and output, a system-level flowchart of each module (ELIPSE and ASSIGN), a description of the table storage format, two examples of the lookup procedure, a summary of ELIPSE output options, a feature-space diagram illustrating a cross-section of the geometry represented by three tables, and a description of NTRAN (one page illegible) and the FLD function. However, there were no user instructions, and no definitions of input quantities. Leaving all of the ELIPSE output options "on" expended several hundred pages of printout and several minutes of computer time before any results were produced. It is likely that the quantity most troublesome to the unfamiliar user will be the quadratic threshold value Q. It seems to be necessary to develop a "feel" for this quantity to use ELLTAB effectively.

The following results were obtained from test runs: The test area was a 200 x 260 segment of Landsat data. Training statistics for six classes were derived from 100 samples each.

The value of Q corresponding to excluding an average of 100 points from each of the six classes (an exclusion probability of 0.01154) is $Q = 12.96$. Using this value for each class, the run of ELIPSE to make a table tape took 0.8 minute (CPU time). The run of ASSIGN, classifying 52,000 points, took a little less than 0.5 minute. This time included reading data cards, reading the data tape, unpacking, classifying, packing, and writing the output tape. The time for classification alone was 250-300 microseconds per pixel. This time should be regarded as typical; however, it could vary in other cases. Probably large homogeneous areas could be classified faster than regions where there are frequent changes between classes. Classification time should increase with the number of classes (as is the case with other classification programs). It should be emphasized that these times were measured on a UNIVAC 1108, and cannot be compared directly with running times on another computer. It should also be emphasized that, although ELLTAB is restricted to Landsat tapes, this lack of generality is compensated for by speed. It is not necessary to precede a run of ELLTAB with a run to unpack and reformat the data.

For Q=12.96, the expected number of points not assigned to any class was 600. The actual number was 1220.

In the table-generation step of this test, the values input for "minimum and maximum scanner output values" were the extreme values actually present in this data set, as determined from a histogram of the occurrence of data values. This reduced the sizes of the tables generated by ELIPSE. Since the tables occupy a significant amount of storage, the simple preliminary step of making a histogram would appear to be worth the trouble.

It would seem that the Gaussian maximum likelihood classification rule without quadratic thresholds is equivalent to the method with quadratic thresholds having arbitrarily (or at least sufficiently) large values, therefore small probabilities of exclusion. However, results from ELLTAB did not agree with this expectation. For Q = 18.16 (exclusion probability 0.001154), the tables for three of the six classes were larger than for Q = 12.96, but the tables for the other three were much smaller. (Table size is related to the size of the corresponding ellipsoid. It should be recalled that in regions of overlap in feature space class assignments are made on the basis of relative assignment probability.) No points--not even the training samples for these classes--were assigned to the latter three classes, and 31,616 points were not assigned to any class.

Results were even worse for an exclusion probability an additional order of magnitude smaller (Q = 23.30). In generating the table for one of the classes (the one having the largest table for both other values of Q), the expression giving the range of values in one channel involved the square root of a (significantly) negative number.

These failures may indicate program bugs or inaccuracies in some of the calculations. At present, a qualitative evaluation is that good results seem to be obtained for values of Q corresponding to exclusion probabilities of the order of 1 percent, but unreliable results are produced for significantly larger Q values.

Further investigation has revealed one problem area. The most complicated calculations in ELLTAB are in the table generation phase. The range calculations for the tables (e.g., the range of X_4 for each possible X_1, X_2, X_3 combination) involve the solution of quadratic equations whose coefficients are complicated expressions. It was found that the implementation used for those equations sometimes led to loss of precision. Changing those equations from single to double precision was found to lead to significant differences in some cases. A recheck was made of the Q = 12.96 results; there were no changes in the tables for that case. However, it is likely that the anomalous results for larger values of Q were influenced by this problem. In particular, the situation described above for the Q = 23.30 calculation was the same as was corrected in other tests by the change to a double-precision calculation.

5.0 LINEAR CLASSIFIER

The linear classifier described here is a supervised, nonparametric technique. Thus, the initial phase of the classification process consists of the definition of a set of discriminant functions using data samples whose classifications are known.

In separating one class of objects from one or more other classes, it is desirable to de-emphasize the characteristic features that the classes may have in common, and to emphasize where possible the features that are unique to the class of interest. The most obvious first approach is to say that the distinctive character of an object or class of objects is the sum total of its features, some features being more distinctive than others in certain environments. The Linear Classifier concept depends upon this assumption, and aims at developing a single measure of a class's composite features. This measure, the discriminant, is formed by adding the value of each feature (reflectance value or brightness in the case of multiband imagery), after each feature has been weighted according to its usefulness in separating the class of interest from the other classes.

5.1 RESOURCE REQUIREMENT

The linear classification system requires as input the number of classes and spectral bands in the data and a set of training samples.

The data samples to be classified are assumed to be arranged in feature vectors on a FORTRAN-readable input device, and the classification results are written on an output tape.

The linear classifier package requires the following arrays:

- an array to store the training samples by class number, feature number, and sample number,

- an array into which the input feature vectors are read,

- an array into which the output class numbers are placed, to be written as output,

- arrays to store the discriminant coefficients by class, and the order in which discriminants are tested,

- a work array for storing the interclass and intraclass distances for each feature, and

- two work arrays each dimensioned according to the number of training samples, for use in training.

The program subroutines required to perform various tasks in a linear classification system are given in Table 5-1, along with their storage requirements. These requirements are constant, as storage for the problem-dependent arrays listed previously is allocated in the main control program.

Table 5-1. Linear Classifier

Subroutine	Purpose	360/65 Storage (8-bit bytes)
EFFECT	Compute interclass and intraclass distances; determine order of class separability	3.2×10^3
SNOPAL	Supervised nonparametric learning of linear discriminant functions	4.1×10^3
NTEST	Test classification of training samples	1.4×10^3
NCLASS	Handle I/O to classify unknown data sets; compute class occupancies	1.3×10^3
NOPACA	Linear discriminant classification of a feature vector	6.2×10^2
GASINV	Matrix inversion; required in learning algorithm	1.8×10^3
SORTSL	Sorting algorithm; required in class ordering	1.1×10^3

5.2 ANALYSIS PROCESS

Nonparametric methods are so termed because the parameters of the distribution functions of the data are not used. The training algorithm determines the values of the weighting factors "w" to be used in a discriminant function of the form

$$G = w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n$$

A set of weights is determined for each class of data, the value of a weight reflecting the significance of its associated feature in separating the class from its companion classes. Thus for each unknown feature vector, a value of G is obtained for each class.

There are two approaches possible in the application of linear classifiers. In the first, the discriminant functions are designed such that one class may be separated from each of the other classes, pairwise. Then, in determining the class to which a particular feature vector (the reflectance values from one pixel) should be assigned, the value of G is calculated by substituting the values of the feature vector in the discriminant function for each of the classes. The class for which the value of G is largest is the class to which the feature vector is assigned.

In the second approach, the one employed at NASA-MSFC,* the discriminant functions are designed such that one class may be separated from all of the other classes considered collectively as one class. Unlike the first approach in which all discriminants are calculated concurrently, here the discriminants are calculated sequentially. Referring to Figure 5-1, the straight line corresponds to the discriminant function that will separate Class 4 from Classes 1, 2, and 3 taken together. If a given feature vector lies to the right of this line, the discriminant has a positive value and the vector is assigned to Class 4. If it lies to the left of the line, the discriminant has a negative value, and the vector is not assigned to Class 4. Class 4 may then be removed from consideration, and a further test is applied using the discriminant function for Class 3, say. These tests are repeated until the feature vector is assigned to a particular class, at which time testing ceases, and a new unknown feature vector is called in. The sequential nature of testing results in a speed advantage over the parallel procedure employed in the first approach.

*"An Integrated Feature Selection and Supervised Learning Scheme for Fast Computer Classification of Multi-Spectral Data," A. D. Bond and R. J. Atkinson, Remote Sensing of Earth Resources, Vol. 1, F. Shahrokhi, Ed., U. of Tenn. Space Institute, March 1972.

The linear classification scheme described here is combined with a feature selection algorithm that determines which of the features of any class are of greatest significance in separating that class from the others. The method of feature selection is based on the concept that the classification is more accurate if

- data values from different classes are widely separated (interclass distance is large), and
- data values within each class are closely grouped (intraclass distance is small).

These effects are illustrated in Figure 5-2.

The interclass and intraclass distances are computed for each feature by calculating the totais of the separations between all pairs of points in different classes (interclass) and within each class (intraclass). The optimum is obtained when the interclass distance is maximized and the intraclass distance is minimized.

After calculating the criterion for best features (based on separations between training data of the various classes), the feature selection values are combined to yield a value which determines the most easily separable class (Class 4 in Figure 5-1), for which the discriminant function coefficients (w's) are then computed.

The analysis process in the training phase is illustrated in Figure 5-3. After the training samples have been selected, they are processed by the feature selection algorithm EFFECT. This determines which class is the most easily separable from all others, and the optimum subset of features (spectral bands) for separating that class. This latter option may be bypassed if not many (three or four for example) spectral bands of data are available, but it is very useful if many bands of multispectral scanner data have been acquired. The discriminant weights for the most easily separable class are then calculated, using the algorithm SNOPAL.

The values of the weights are determined by an iterative procedure. In each iteration, the value of w is changed slightly from its previous value to produce an improved set of weights. Several options are available in the algorithm for terminating the iteration. Once the weights for the most easily separable class have been determined, the training samples for that class are removed from the data set, and EFFECT then determines the next most easily separable class and its optimum feature subset. Then SNOPAL computes the required discriminant function coefficients. This process of identifying an easily separable class and its discriminant, suppressing its data and moving on to the next easily separable class, is repeated until a discriminant function has been calculated for all of the classes in the training data set.

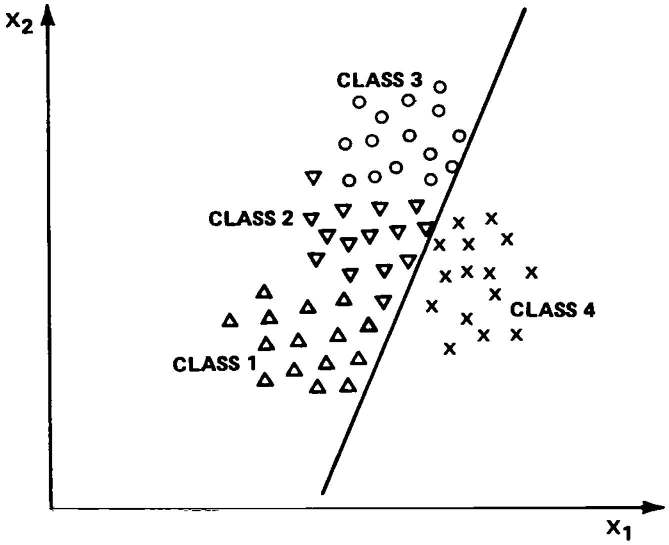
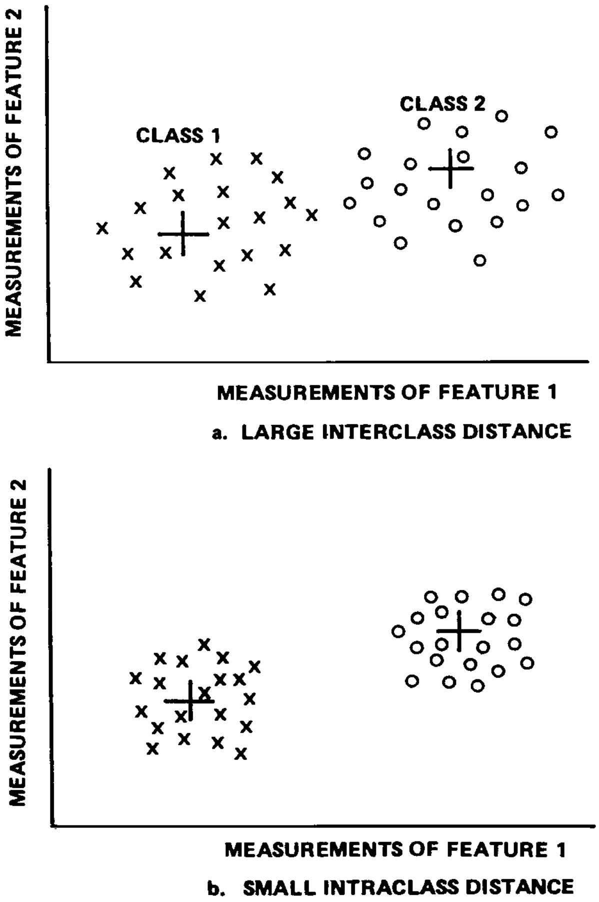


Figure 5-1. Decision Function for Assigning Samples to Class 4



a. LARGE INTERCLASS DISTANCE

b. SMALL INTRACLASS DISTANCE

Figure 5-2. Interclass and Intraclass Distances

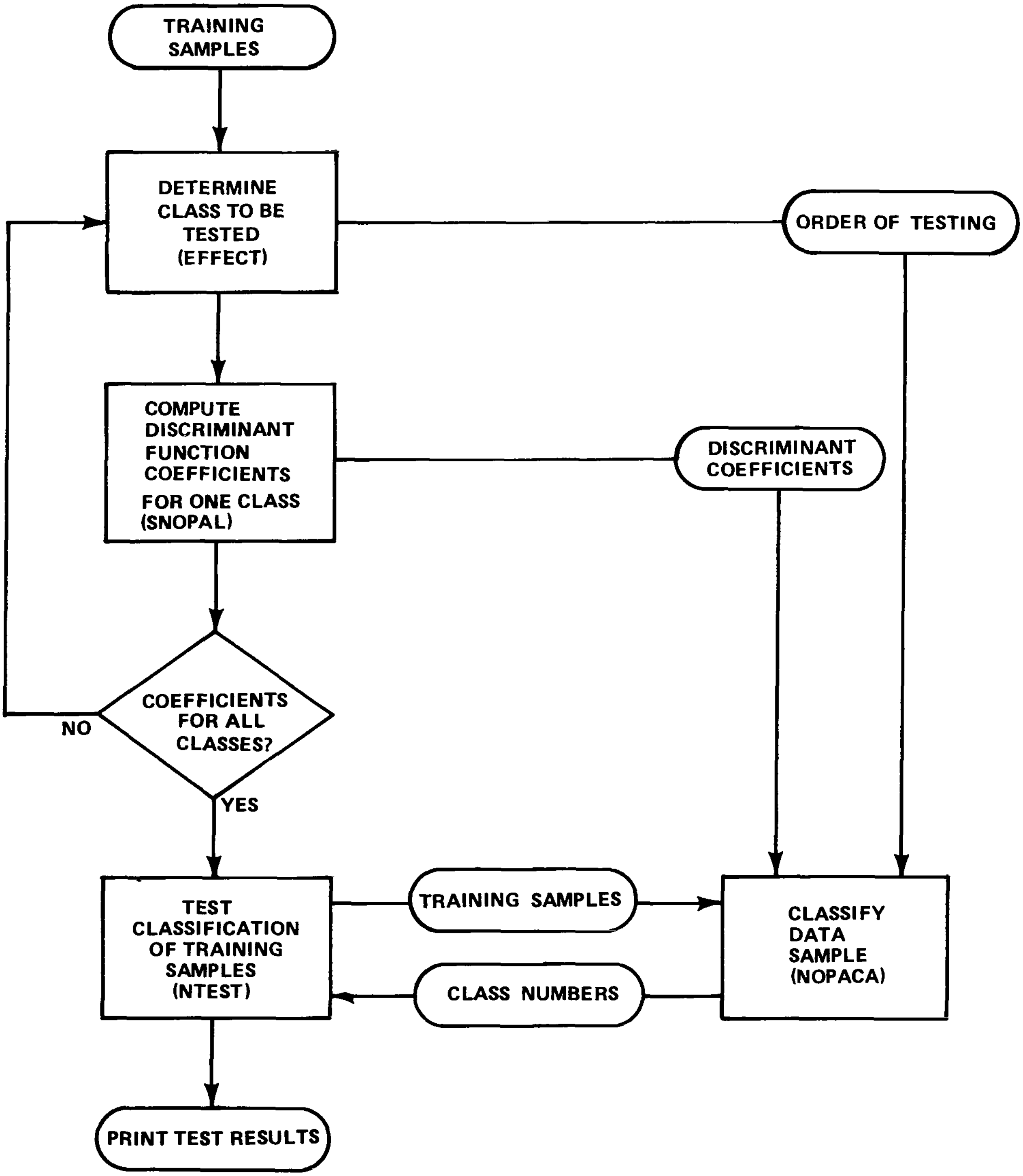

TRAINING SAMPLES
DETERMINE CLASS TO BE TESTED (EFFECT)
ORDER OF TESTING
COMPUTE DISCRIMINANT FUNCTION COEFFICIENTS FOR ONE CLASS (SNOPAL)
DISCRIMINANT COEFFICIENTS
COEFFICIENTS FOR ALL CLASSES?
NO
YES
TEST CLASSIFICATION OF TRAINING SAMPLES (NTEST)
TRAINING SAMPLES
CLASS NUMBERS
CLASSIFY DATA SAMPLE (NOPACA)
PRINT TEST RESULTS

Figure 5-3. Discriminant Training Phase of Linear Classifier

The training phase is completed by performing a test classification of all training samples. Ideally, the classifier should assign the training samples to the class from which they were selected by photointerpretation. If the classifier assigns more than a few samples from Class 4 to Class 1, for example, this will suggest an unsatisfactory choice of training samples, and that some of Class 4's training samples were inadvertently selected from Class 1. The choice of training samples must then be revised, and the entire training phase repeated.

In the classification process for an unknown feature vector, shown in Figure 5-4, the values "G" of the discriminant functions are computed in the same order as the functions were defined, and the assignment is made to that class for which G is a positive number.

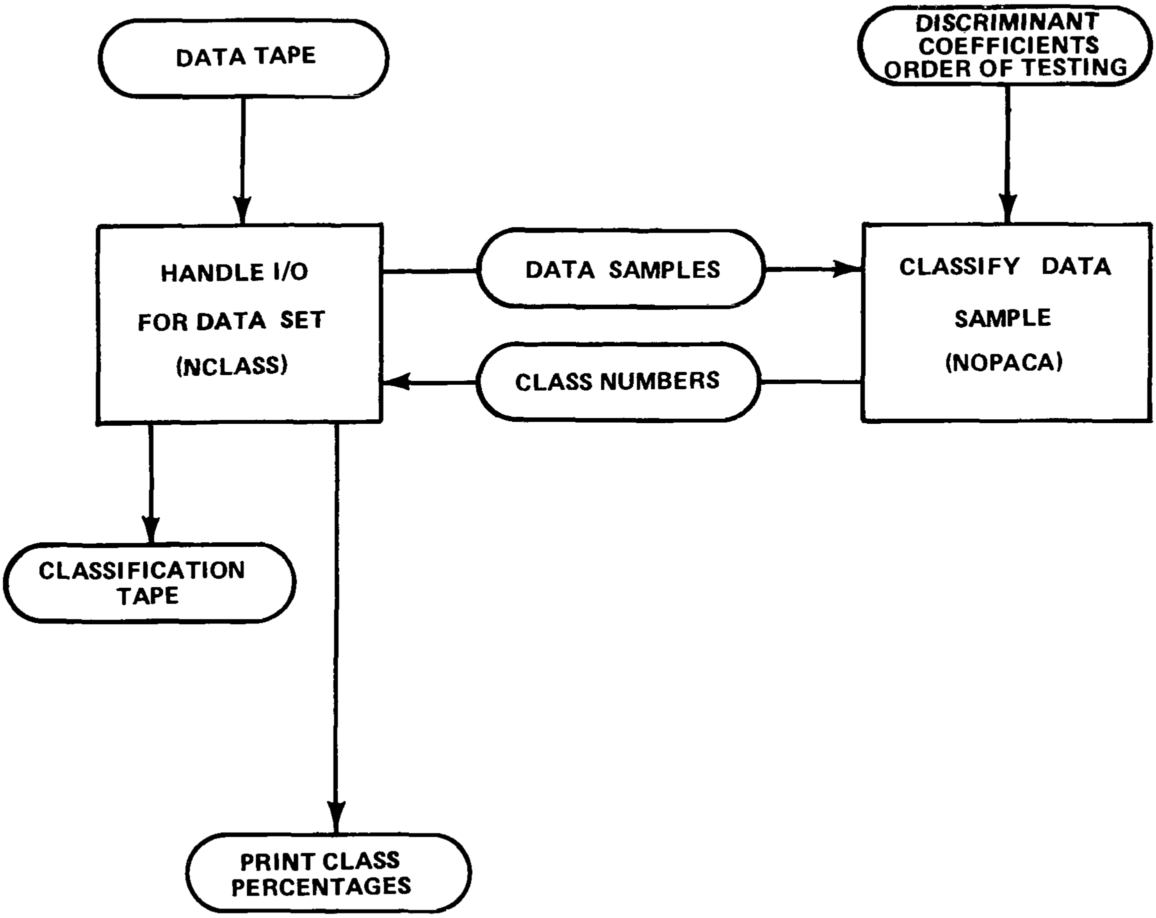

DATA TAPE
DISCRIMINANT COEFFICIENTS ORDER OF TESTING
HANDLE I/O FOR DATA SET (NCLASS)
DATA SAMPLES
CLASSIFY DATA SAMPLE (NOPACA)
CLASS NUMBERS
CLASSIFICATION TAPE
PRINT CLASS PERCENTAGES

Figure 5-4. Classification Phase of Linear Classifier

5.3 PERFORMANCE CHARACTERISTICS

This method will operate satisfactorily on large numbers of spectral bands and classes. The size of the data set to be classified is immaterial.

The output of the classification program is a tape containing the classification map and a listing of the class populations and percentages.

Table 5-2 gives classification rates for various data sets. Computer time required to read and write the input and classification tapes is not included. Although one normally expects the classification time of the linear classifier to increase in direct proportion to the number of spectral bands analyzed, reference to the third and fifth entries of the table shows this not to be a general rule of thumb. Here the processing time increased by a factor slightly more than 2 although the number of bands increased by a factor of 4. This apparent anomaly can be accounted for partly by a certain factor due to overhead in the computer system software, and partly by differences in the sequential operation of the NOPACA algorithm when processing different data sets.

Table 5-2. Linear Classifier Times

Number of Classes	Number of Spectral Bands	Number of Data Points	Computer Time Seconds	Points Classified Per Second
3	4	510,000	129	3981
4	4	3,750,000	1279	2933
6	4	52,000	18	2970
7	4	3,750,000	2006	1869
6	16	52,000	39	1337

6.0 BINARY CLASSIFIER

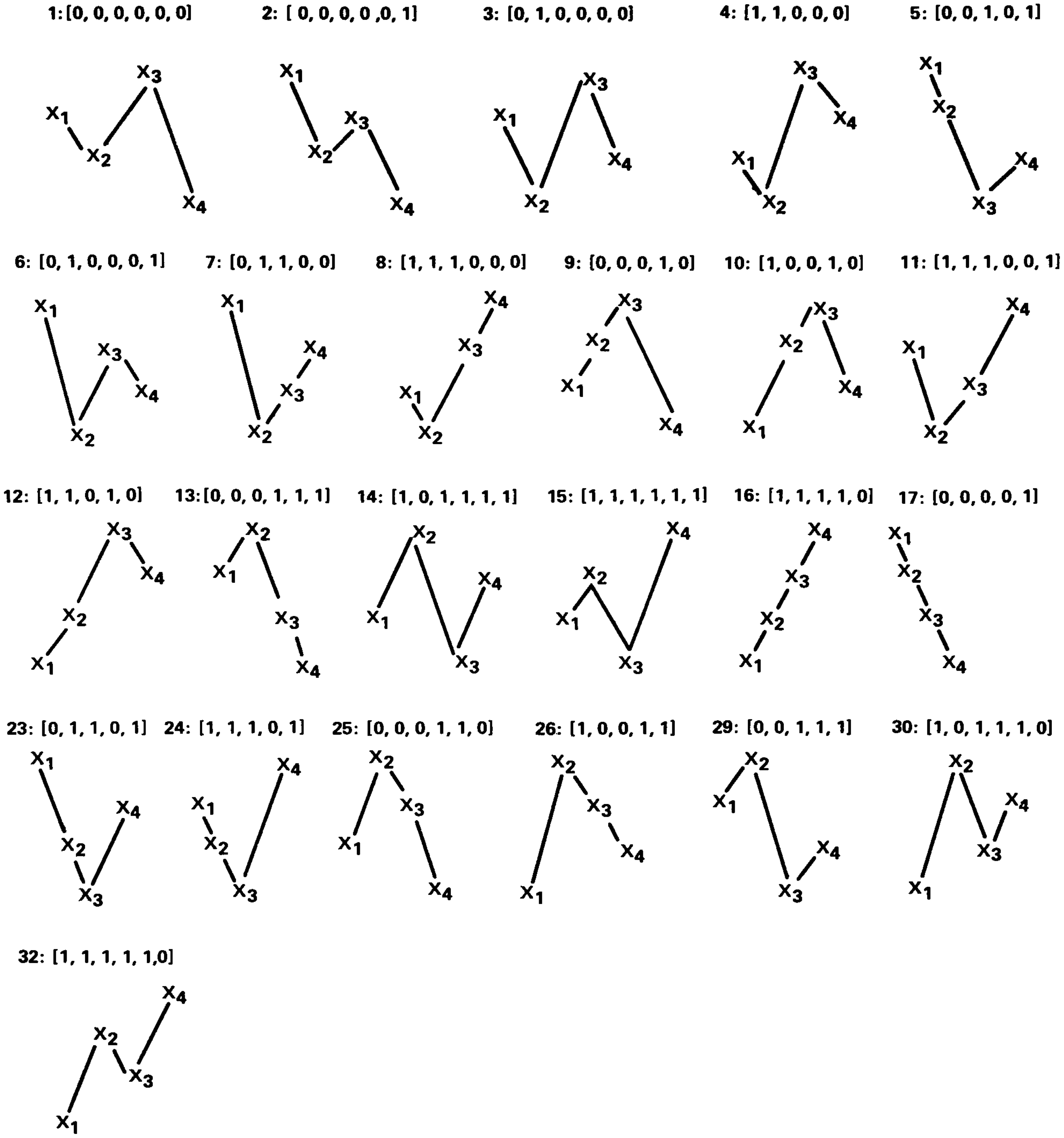
The Binary Classifier is an unsupervised classification program specifically designed for Landsat data that extracts a maximum of 24 classes. The classification scheme is based on the shape (amplitude ratios) of the four channel vectors of which there are 4! or 24 different possibilities. The decision spaces for the different classes all have a common intersection at a line whose direction cosines are [1/2, 1/2, 1/2, 1/2] and, hence, the decision spaces lie in rotation about this line. The program compresses the 4-channel Landsat data into a single band image containing a maximum of 24 different integers.

6.1 RESOURCE REQUIREMENTS

The program is utilized in much the same manner as a program to reformat a data tape, and therefore requires only an input and output tape and no prior knowledge of the features contained in the data set. Hence there are no input parameters that control the classification process. Also, the program currently uses 104K 8-bit bytes of core memory, much of which is unnecessary. There are no documentation or results for this program except for what is contained in this report.

6.2 ANALYSIS PROCESS

For a particular pixel, let x_i be the value of the data in channel i. The program creates a binary vector for each pixel by comparing x_4 with x_1, x_4 with x_2, x_4 with x_3, x_2 with x_1, and x_2 with x_3. The binary vector contains a component for each comparison which is either a zero or a one. For example, if $x_4 \geq x_1$, the first component is one, otherwise it is zero. The second component is determined in a similar manner by comparing whether or not $x_4 \geq x_2$. The third, fourth, and fifth components are determined from $x_4 \geq x_3$, $x_2 \geq x_1$, and $x_2 \geq x_3$, respectively. This binary vector is then converted to a decimal number by dotting it with a vector whose components are $[2^0, 2^1, 2^2, 2^3, 2^4]$. As a result of the dot product, it is possible to generate the following 18 decimal numbers: 1, 3, 4, 5, 7, 8, 9, 10, 12, 16, 17, 23, 24, 25, 26, 29, 30, and 32. There are, however, six decimal numbers (1, 3, 8, 25, 30, and 32) which represent two differently shaped feature vectors, and these vectors can only be distinguished by going to a sixth comparison, $x_1 \geq x_3$. Thus, if $x_1 \geq x_3$ for the decimal numbers 1, 3, 8, 25, 30, and 32, then 1 is changed to 2, 3 to 6, 8 to 11, 25 to 13, 30 to 14, and 32 to 15 by assignment. The 24 possible classes are indicated by the following decimal numbers: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 23, 24, 25, 26, 29, 30, and 32. Figure 6-1 shows the possible four



1:[0, 0, 0, 0, 0, 0]
2: [0, 0, 0, 0 ,0, 1]
3: [0, 1, 0, 0, 0, 0]
4: [1, 1, 0, 0, 0]
5: [0, 0, 1, 0, 1]
6: [0, 1, 0, 0, 0, 1]
7: [0, 1, 1, 0, 0]
8: [1, 1, 1, 0, 0, 0]
9: [0, 0, 0, 1, 0]
10: [1, 0, 0, 1, 0]
11: [1, 1, 1, 0, 0, 1]
12: [1, 1, 0, 1, 0]
13:[0, 0, 0, 1, 1, 1]
14: [1, 0, 1, 1, 1, 1]
15: [1, 1, 1, 1, 1, 1]
16: [1, 1, 1, 1, 0]
17: [0, 0, 0, 0, 1]
23: [0, 1, 1, 0, 1]
24: [1, 1, 1, 0, 1]
25: [0, 0, 0, 1, 1, 0]
26: [1, 0, 0, 1, 1]
29: [0, 0, 1, 1, 1]
30: [1, 0, 1, 1, 1, 0]
32: [1, 1, 1, 1, 1,0]

Figure 6-1. Four Channel Vector Shapes

channel vector shapes, the decimal class number and the binary vector, whose sixth component is indicated when appropriate.

6.3 PERFORMANCE CHARACTERISTICS

The program is dimensioned to handle four channels of data, a total of 824 samples per channel per scan (record), and as many scans of data as required. The maximum number of classes is 24. The outputs of the Binary Classifier are a table indicating class population and percentage and a tape whose information is normally converted to a photographic product. The results of this program need to be further evaluated for the effects of preprocessing such as calibration and density stretching.

7.0 SPATIAL AND SPECTRAL CLUSTERING PROGRAM (SSCP)

The Spatial and Spectral Clustering Program (SSCP) can be run in either an unsupervised or supervised mode, and is composed of two modules which are run separately. The first module allows a user to select training areas manually or will automatically select training areas based upon the spatial and spectral characteristics of the data set, and automatically merges data from training areas that are spectrally similar. The second module classifies each individual pixel according to whether or not it belongs to one of the described classes. Each class is described by a mean vector and a set of eigenvectors and eigenvalues, which are derived from module one and used in module two. The classification is thresholded, which usually results in some pixels remaining unclassified.

7.1 RESOURCE REQUIREMENTS

The resources required by a user, as far as a knowledge of the data set is concerned, can range from very little knowledge to considerable detailed information, since the program can be run in either a supervised or unsupervised mode. The resources that are available concerning a programmer's documentation of the program are fair to poor, but the transfer of the program to a user can be accomplished. There are, however, two reports that mathematically describe the program and present results on aircraft scanner and Landsat multispectral data.*

The program will operate on single channel as well as registered multispectral data, and the only specialized routine (developed specifically for the IBM-360/65) that is separate from the program but needed, is a routine to reformat the data. The program (SSCP) accepts data in the format of one scan being a record and each pixel (picture element) being represented by a vector, whose components are the amplitude of the data in each channel.

The program, as it is currently used, is run in two parts. The first part acquires the statistics necessary to classify the data and uses 206K eight-bit bytes of core memory. This part of the program also utilizes four tape drives and eight sections of disc storage, each of which contains 2341 blocks (records) of 1028 bytes. One

*"Unsupervised Spatial Clustering with Spectral Discrimination," R. R. Jayroe, NASA TN D-7312, May 1973.

"Computer and Photogrammetric General Land Use Study of Central North Alabama," R. R. Jayroe, P. A. Larsen, and C. W. Campbell, NASA TR R-431, October 1974.

of the tapes contains previously acquired statistics, if there are any, the second tape contains the reformatted data, and the third tape contains the output statistics used in classifying the data. The fourtb tape is optional and contains the cluster map.

The second part of the program classifies the individual pixels based upon the acquired statistics and utilizes 110K eight-bit bytes of core memory. This part of the program also utilizes three tapes which contain the input statistics, the input reformatted data, and the output classification map. One section of disk is reserved that contains 2340 blocks of 3300 bytes.

There are a total of seven different parameters, which have a dominant effect on the accuracy of the classification, and these will be discussed in the next section.

7.2 ANALYSIS PROCESS

The program contains two modules which are presently run separately. The first module performs three different operations on the data, while the second module only classifies the data. Thus, the entire program consists of a boundary routine, a spatial clustering routine, a spectral merging routine, and a classification routine.

The purpose of the boundary routine is to compress the multichannel data into one channel of data for the spatial clustering and at the same time categorize the data into spatially and spectrally homogeneous areas that are separated by boundaries. This approach provides a computer map similar to what would be obtained by a draftsman tracing a map from a photograph. Mathematically, the boundary map is produced in the following manner. Each pixel of data is represented by a multispectral vector whose components are the amplitude of the data contained in the different channels. The spectral vector distance is computed between the pixel in question and the previous adjacent pixel in the same scan, and also between the pixel in question and the adjacent pixel in the same column, but in the previous scan. If the vector distances are large enough, this indicates spectrally that a new or different feature is being encountered in the data. Such a large change occurring between adjacent pixels in the same scan would indicate a vertical boundary, while a large change occurring between adjacent pixels in the same column would indicate a horizontal boundary. A boundary map of the data set is then produced showing where these large changes or boundary pixels occur.

The spatial clustering routine uses the boundary map as an input and searches the boundary map for homogeneous areas. This search is accomplished by using a fixed shape array (maximum size is 11 samples wide by 11 samples long) that queries the boundary map. The rules for the array are that it cannot enter the boundary map if it will contain a boundary point, and the array can move to the right and down until it encounters a boundary point once it has entered the boundary map. Each time the array enters the boundary map, a new cluster is started and all of the locations consumed by the array are given the same number on the boundary map, indicating which spatial cluster they belong to. If two spatial clusters overlap by four or more scans, they are spatially merged and defined to be the same cluster. Thus, the boundary mapping and spatial clustering is a machine analog to selecting training areas manually. The user could bypass this much of the program by manually inputting the coordinates of desired training areas.

The purpose of the spectral merging routine is to determine which spatial clusters are spectrally similar and which ones are spectrally distinct. The inputs to this routine are the raw data and the cluster map or training area coordinates, which provide the program with information on where to fetch the raw data for each cluster. Once the data have been fetched, the following quantities are calculated for each cluster:

- pixel population,
- mean value for each channel (i.e., mean vector),
- covariance matrix,
- eigenvectors, and
- eigenvalues.

The data belonging to each cluster are then enclosed by a surface in the multispectral space whose dimension is equal to the number of channels of data. This closed surface is a hyperellipse whose center of location is the mean vector, whose orientation is given by the eigenvectors, and whose extent in the direction of orientation is governed by the magnitude of the eigenvalue associated with its eigenvector. The rule for spectrally merging two clusters is that the mean vector of each cluster must be contained in the other cluster's closed surface. When two or more clusters are spectrally merged, the previously mentioned quantities are recalculated for the combined data of the merged clusters. Once the merging process has been completed, the remaining clusters are called classes.

The classification program then classifies each pixel as to whether it belongs to a particular class or none of the classes. The rule for classification is that the pixel vector first must be contained within the closed surface defining a class and, secondly, if it is contained within more than one such surface, the pixel vector is assigned to the class whose center location (mean vector) is the closest. The inputs to the classification program are the raw data and the class statistics, which are the mean vectors, eigenvalues, and eigenvectors.

If the classification map is incompletely classified, the clustering and merging program can be resubmitted on the unclassified areas and a new updated statistics tape will be created. This procedure can be repeated as many times as desired.

A flowchart of the program is shown in Figure 7-1 along with where the parameters are used that affect the classification accuracy. The definitions and use of these parameters are as follows:

MANUAL — When MANUAL=0, the clustering program will pick training areas from the boundary map. When MANUAL=1, the user can select up to 46 training areas manually by using the parameter ISELET.

ISELET — Input coordinates for the training areas that are manually selected. For each training area, six coordinates are input in the following manner: start scan, stop scan, start scan start column, stop scan start column, start scan stop column, and stop scan stop column.

IXXX — The width in the data columns of the fixed shaped clustering array used on the boundary map when MANUAL=-. IXXX ≤ 11.

IYYY — The length in data scans of the fixed shaped clustering array used on the boundary map when MANUAL=0. IYYY ≤ 11.

NCLUST — NCLUST is equal to the number of sets of class statistics contained on the input statistics tape for both the clustering and the classification programs.

SCLMRG — This parameter controls the extent of the closed surface for all clusters used in the clustering program and hence governs whether or not clusters will merge. More clusters will merge together when the value of SCLMRG is made larger. For four channels of Landsat data, SCLMRG is normally equal to 1.

SCLCLS — This parameter controls the extent of the closed surface for all classes used in the classification program for classifying data vectors. As SCLCLS is made larger, more data vectors will be included in each class. For four channels of Landsat data, SCLCLS is normally equal to 2.25.

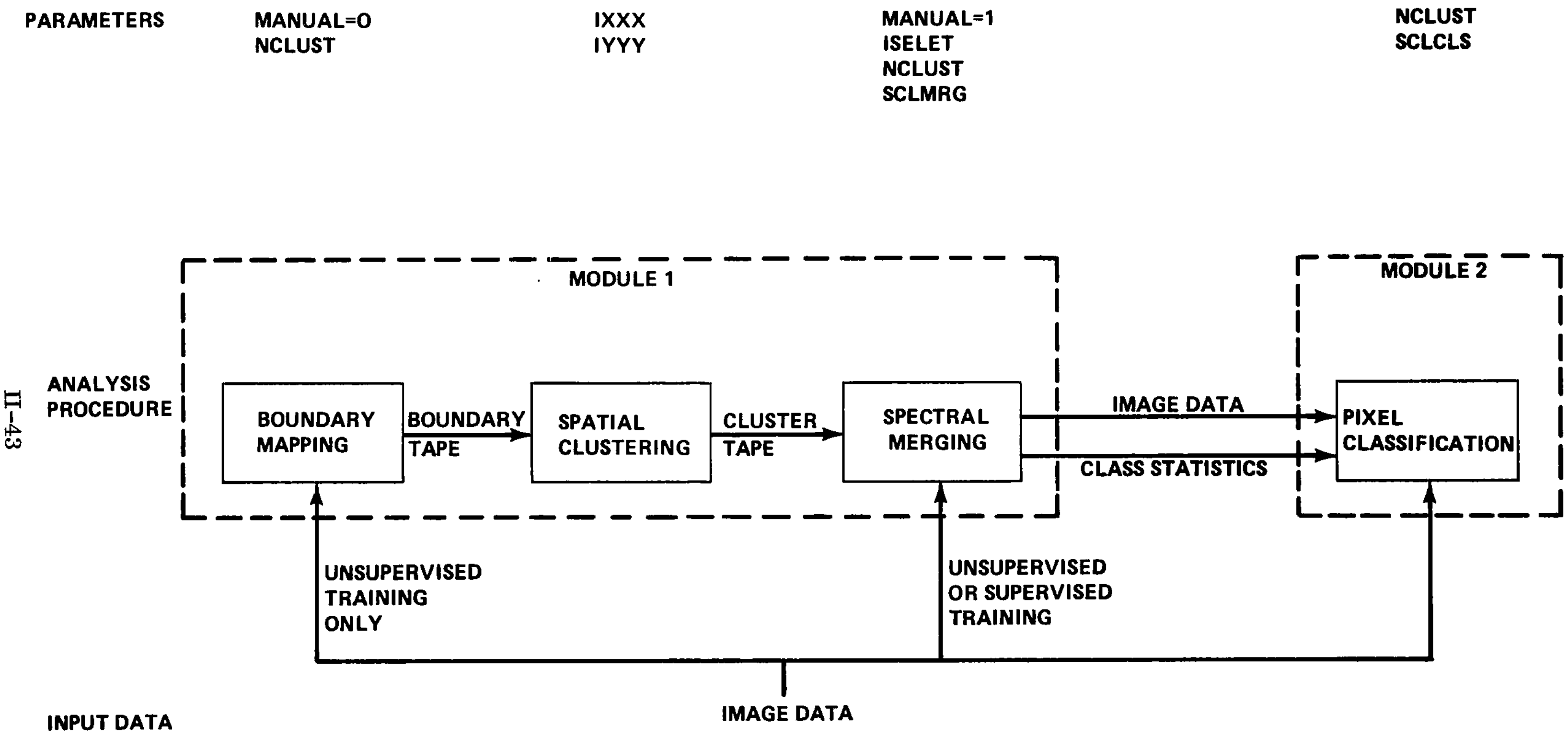


Figure 7-1. Flowchart of Spatial and Spectral Clustering Program (SSCP)

7.3 PERFORMANCE CHARACTERISTICS

SSCP is dimensioned to handle a maximum of 12 channels of data, 150 unmerged clusters, or 46 input training areas, and 42 final classes. The clustering part of the program is dimensioned to handle a strip of data 256 columns wide and as many scans as desired, while the classification program is dimensioned to handle 824 columns of data and as many scans as desired.

In the unsupervised mode, the program tends to work best on large data sets where there is the opportunity to find large (typically 10x10 pixel arrays) homogeneous areas on the boundary map. The most critical part of the analysis is making sure that the boundary map contains enough boundaries so that data from different ground scene features do not spatially mix and that SCLMRG is small enough so that different features are not spectrally mixed together. The analysis can be checked by examining the printout of the clustering program which contains the following:

- a boundary map showing the location of each cluster,
- the statistics associated with each cluster (population, mean vector, covariance matrix, eigenvectors, and eigenvalues),
- the cluster number and the cluster or clusters that were merged together as well as the updated statistics, and
- the final class assignment of each cluster and final statistics for each class.

The output of the classification program is a tape containing the classification map and a listing of the class population and percentages. A separate program is used to obtain a printout of desired portions of the map, but the data on the tape is normally converted to a photographic product. Typically, it is possible to classify at least 90 percent of the data using the program in the unsupervised mode. The urban category is usually the most difficult to classify using the unsupervised mode because urban areas tend to end up as boundaries.

Typical 360/65 running times on previously analyzed Landsat data sets are shown in Tables 7-1 and 7-2. The column entitled "Total Cluster Population" in Table 7-1 is the percentage of the data set that was used in calculating statistics for all of the clusters, while the column entitled "Classification Percent" in Table 7-2 is the percentage of the data set that was classified into one of the permissible classes. The remaining data was unclassified. In the clustering program, the length of time required to produce a boundary map is directly proportional to the number of pixels in the data set or about 335 pixels per second. There appears to be no

Table 7-1. Clustering Program Running Times

No. of Pixels	Boundary CPU Time	No. of Clusters	Total Cluster Population	No. of Classes	Clustering CPU Time
120,000	5 min., 58 sec.	49	6.65%	13	11 min., 09 sec.
120,000	5 min., 58 sec.	56	23.33%	13	11 min., 10 sec.
264,000	13 min., 07 sec.	52	71.36%	11	28 min., 04 sec.
288,000	14 min., 18 sec.	45	82.56%	9	20 min., 17 sec.
328,000	16 min., 17 sec.	139	37.26%	9	29 min., 14 sec.

Table 7-2. Classification Program Running Times

Number of Pixels	Number of Classes	Classification Percent	Classification CPU Time
451,000	11	86.85	22 min., 47 sec.
451,000	11	91.52	21 min., 55 sec.
891,000	11	90.16	38 min., 29 sec.
1,223,100	9	79.60	46 min., 42 sec.
1,223,100	9	83.21	46 min., 00 sec.
1,223,100	13	69.18	67 min., 11 sec.
1,223,100	13	83.48	62 min., 31 sec.
1,223,100	13	90.65	58 min., 12 sec.
1,223,100	13	91.61	56 min., 47 sec.

simple time relationship involved with the clustering except that it, too, tends to increase with the number of pixels. The classification program time also appears to be linear with the number of pixels for a given number of classes, except for a variation due to the percent of the data classified. This variation is caused by the classification logic which checks the class assignment made to pixels spatially adjacent to the pixel in question. If the pixel in question will fit into an adjacent pixel class, there is no need to check for the other possible class assignments. However, if a spatially adjacent pixel is unclassified, all possible class assignments have to be checked for the pixel in question. Hence, fewer possible class assignments have to be checked as more pixels are classified. The main currently recognized bottleneck in the program, which concerns running time, is the way that the data is read into the program. The data is read into the program from tape by a subroutine which is not efficient on the IBM-360/65. Since the running time is highly dependent on the number of pixels, it is anticipated that a significant reduction in running time can be achieved by rewriting the data reading subroutine.

8.0 HISTOGRAM INSPIRED NEIGHBORHOOD DISCERNING UNSUPERVISED (HINDU) SYSTEM

This technique comes under the category of unsupervised, nonparametric classification techniques and is most suited for application to environments wherein neither ground truth nor information about the distributions underlying the data are available. The methodology is highly automated and requires little human interaction. User's subjective influence on the process is limited to prescription of the maximum and minimum limits on the number of clusters. The only other input parameters that need to be specified, in addition to these subjectively chosen maximum and minimum limits, are the size of the data set in terms of dimensionality of the measurement vector, number of measurement vectors, and the approximate range in the values of measurements. Assuming the data to be available on tape, the system expects to have the data in measurement vector format with all the feature values of each data point input consecutively and accordingly the size of the data set is to be specified by the number of data points per record and the number of such records. With this input of unlabeled data set, the HINDU system derives the corresponding output label set.with no further human intervention.

8.1 RESOURCE REQUIREMENTS

The program, as is currently implemented, is dimensioned to handle four dimensional data sets. There is no critical limitation on the number of scan lines and the number of data points/scan line, and accordingly there is no strict limitation on the size of the data set. A typical setup of up to 500 data points/scan line calls for a core memory requirement of 150K (8-bit) bytes. For input/output of data sets and labels, two tape drives are called for by the program (one of the tape drives can be substituted for by a disk depending on the user's resources for display of results, etc.). The only other requirement external to the program is a matrix inversion subroutine. There are no other special processing requirements.

The input parameters that need to be specified by the user are:

N:	Dimensionality of the data set
MS:	An estimate of the maximum spread in the data values
IS:	Number of data points/scan line
NSL:	Number of scan lines of data
MAX:	Maximum number of clusters acceptable to the user
MIN:	Minimum number of clusters acceptable to the user

CPM: Cluster population minimum (may be set to zero if unknown)

IC: Initial grid size for the histogram (may be set to zero if unknown)

CPM is generally set to zero, in which case the program chooses its own threshold for cluster population minimum. Otherwise, CPM is to be prescribed as the smallest cluster population (as a percentage of the total population) deemed significant by the user. Similarly, IC, the initial trial grid size, can be either specified by the user or set to zero (in which case the system internally chooses its own value). The program limitation of $N \leq 4$ can be relaxed by appropriate changes in the dimension statements, but only to a certain extent. Conceptual considerations limit the dimensionality of the data set to relatively modest values and this is discussed under Performance Characteristics (see Section 8.3). The size of the data set is not very critical and the current version of the program handles up to 500 data points/scan line which can be extended if needed by changing the dimension statements. The number of scan lines itself is relatively open ended. The ultimate limitation on the size of the data is, of course, the CPU time and core memory available to the user.

8.2 ANALYSIS PROCESS

The major components of the HINDU system, as shown in Figure 8-1, are:

- Histogram Generator,
- Cluster Formulator,
- Discriminant Designer, and
- Label Designator.

8.2.1 Histogram Generator

The function of the Histogram Generator, as the name implies, is to generate the multidimensional histogram of the input data set. The histogram analysis leads to a set of multidimensional cells occupied by the input data set. The output of this Histogram Generator consists of:

- an address array listing the (multidimensional) address of each of these nonempty cells,
- a density array containing the densities (i.e., number of samples allotted to each) of these cells, and

HISTOGRAM INSPIRED NEIGHBORHOOD DISCERNING UNSUPERVISED (HINDU) SYSTEM

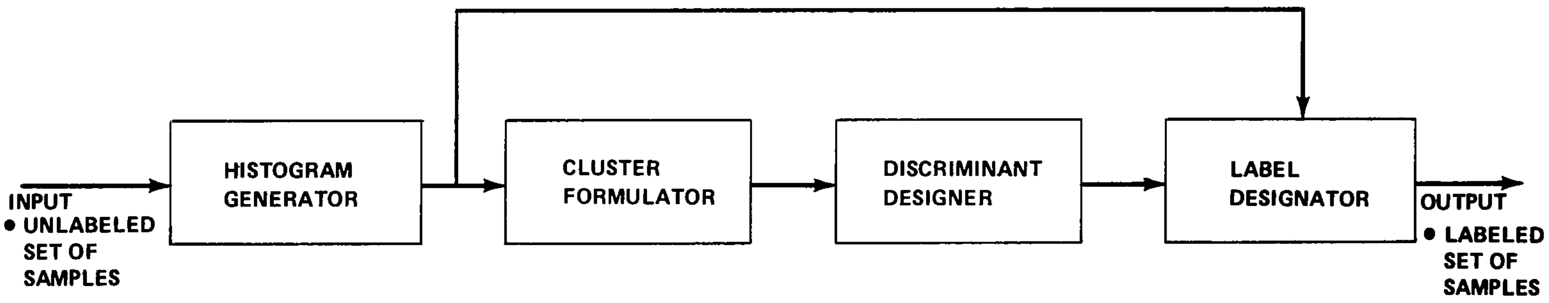


Figure 8-1. Block Diagram of HINDU System

- "n" feature average arrays storing the corresponding feature averages of all the samples contained in each of the cells.

These (n+2) arrays describe in essence the terrain of the histogram in the multidimensional space. The actual length of these arrays (i.e., the number of nonempth cells) depends on the grid size or cell width of the histogram, range (or width) of values in the data set, and the dimensionality of the data. The system is automated so that, depending on the maximum and minimum limits on the number of clusters specified by the user, an appropriate grid size is chosen internally. The system has the flexibility to try alternative grid sizes and choose the one that leads to a permissible number of clusters.

8.2.2 Cluster Formulator

The output of the Histogram Generator is processed here to formulate the cluster (of cells) and define their boundaries. This is achieved by a sequential procedure consisting of the following steps:

- identification of the current lowest density cell,
- connection of this cell to its higher density neighbors by reassignment of the contents of this cell to these neighbors in proportion to their current density levels,
- storage of these connections in memory in the form of a connectivity matrix, and
- updating of the density and average arrays to reflect the changes due to reassignment.

This sequential processing is continued until all the originally nonempty cells are processed. As is to be expected, this processing leads to a finite number of cells whose contents remain unassigned, there being no higher density neighbors to these cells.

These cells are considered as candidate cluster nuclei. However, some of these cells may not truly represent significant clusters, but are merely outliers of the distributions containing insignificant numbers of samples which are possibly just noisy measurements. This can be tackled by establishing a threshold density level (such as average density value of all originally nonempty cells) and considering as significant only those cluster nuclei that have their updated density values higher than the threshold value. Now, this reduced set of nuclei cells represents the cluster cores deemed significant in the given data set.

The connectivity matrix can then be processed to trace out the connections of each cell up to these significant cluster nuclei and thereby identify the clusters of cells surrounding each nucleus cell. Such cells are considered to represent the fuzzy boundary separating the corresponding clusters.

Thus, the Cluster Formulator leads to a set of significant clusters each identified in terms of interior cells (determined as being connected to a single cluster nucleus) and boundaries identified by cells with multinuclei associations.

8.2.3 Discriminant Designer

The objective here is to determine the set of hyperplanes which discriminate between each pair of clusters. The conventional methods of learning the discriminant functions based on error-correcting procedures and solution of linear inequalities are not well suited in view of the fact that there exists a significant amount of information in terms of cells representing the fuzzy boundaries. The methodology adopted here* tackles this modified problem environment by ensuring that the hyperplane represents an optimum fit to the fuzzy boundary in addition to fulfilling its traditional role of being a discriminant between the two identified clusters. This is achieved by viewing it again as a linear inequality problem, but with certain additional minimization constraints and establishing an equivalent unconstrained linear inequality problem amenable to conventional techniques. (Here, the well known Ho-Kashyap algorithm is adopted to handle the equivalent unconstrained linear inequality problem.) This modified method of nonparametric learning of discriminant functions, one of the useful innovations of the system, leads to the determination of the set of discriminant hyperplanes that form the basis of the labeling scheme.

8.2.4 Label Designator

The label designator essentially consists of a table of labels corresponding to the Centroids of the histogram cells as discerned by the Histogram Generator. The identities of the input samples, in terms of their addresses in the histogram space, being known, the labels of the individual samples are derived by looking up this table for the corresponding entries. The labels of the Centroids or the prototypes are of course determined by the discriminant hyperplanes designed earlier. This table lookup approach leads to accelerated recognition and label designation of the input sample set and is recorded onto a tape.

*B. V. Dasarathy, "Discriminant Hyperplane Abstracting Residuals Minimization Algorithm for Separating Clusters with Fuzzy Boundaries," Proc. IEEE, Vol. 64, (to appear in) April 1976.

8.3 PERFORMANCE CHARACTERISTICS

The method is designed for processing relatively large data sets of moderate dimensionality under unsupervised environments wherein computational economy is a significant factor in dictating the choice of the technique to be employed.

This method does not involve intersample distance computations, a common feature of many other clustering approaches, and hence the computational load increases only linearly with increase in data size (and not in proportion to the square of the number of samples as it would be in the other cases). Thus there is not much of a critical limitation in the size of the data set. However, there is a limitation on the dimensionality of the data set because, for a given grid size, the number of occupied cells encountered in the histogram space increases exponentially with increase in dimensionality (of course, with an upper bound in terms of the number of samples in the data set). This increase can be compensated to an extent by increasing the grid size. But the increase in grid size cannot be continued indefinitely as at least a minimum number (three or four) of grid divisions along each dimension is necessary to be able to extract information of value in terms of histogram peaks and valleys along the individual feature directions.

In view of this, the system is presently designed to handle up to 4 dimensional data sets. A preprocessor for dimensionality reduction is suggested in cases wherein the dimensionality of the data set is significantly higher. For small increases in the dimensionality (say up to 5 or 6), the program can itself be redimensioned without going in for a preprocessor, as dimensionality reduction necessarily results in loss of available information. But for large dimensionality data sets, this is unavoidable and for this purpose a computationally feasible preprocessor is available as part of the total system.

CPU time requirements of this method, depending on the data set, can vary between 0.5-2 minutes in the range of data sizes tested. Typically, processing a four-dimensional data set consisting of 1/4 million data points required approximately 89 seconds to derive the label set. Of this, 84 seconds were spent in identifying the significant clusters in terms of their centroids, 3 seconds in establishing their boundaries and defining the hyperplanes, and the rest (2 seconds) in deriving the individual sample labels. The major part of the time is thus spent in learning and identifying the clusters inherent in the environment. The time required in identifying the clusters in terms of the centroids and boundaries (i.e., unsupervised learning) which really represents the core of this method, is less than two minutes in most of these cases and is significantly small unlike most other comparable techniques. Here, the time for labeling of the individual samples is of the order of 7 micro secs/sample and this actually decreases with increase in size of the data set as there is a relatively constant effort involved in creating the label table. Thus, the method becomes much more attractive for larger data sets.

SECTION III

TECHNIQUE ASSESSMENT

SECTION III. TECHNIQUE ASSESSMENT

1.0 INTRODUCTION

One of the most important criteria in the evaluation of classification methods is the accuracy with which they assign class designations to each point in the data. The expected accuracies of certain methods can be evaluated in some cases using theoretical techniques by making assumptions on the statistics of the input data. However, this is difficult in many cases and a comparison of the classifier performance on several data sets with known ground truth is more useful.

This section of the Classification Software Technique Assessment describes the results of tests in which particular data sets are classified using the repertoire of methods described in Section II. In time, it is expected that a body of results and conclusions will be developed to result in firm criteria and guidelines for the use of particular techniques in particular types of applications.

The results obtained by analyzing one data set are presented in this issue. This set includes multitemporal data, that is imagery of one test site acquired at different times of year.

For each data set considered, general descriptions of the data sets are followed by details pertaining to the Ground Truth and its preparation for digital processing. The classification results are shown in pictorial form, and accuracy comparisons between each method used and the Ground Truth Map are presented. Joint histograms (similarity matrices) and difference maps are used extensively to find the accuracies and highlight the differences. Accuracy comparisons between the various techniques, taken in pairs, are also presented. Observations and conclusions based on the analyses are itemized and discussed.

For convenience of reference, performance summary tables are presented in a separate chapter.

2.0 THE BALD KNOB, TENNESSEE, QUADRANGLE TEST SITE

The contributors to this section include Mr. John Wilson, Director of Natural Resources, and his staff from the Tennessee State Planning Office who provided the test site, ground truth information, and user assessment. Dr. C. T. N. Paludan, Chief, Earth Resources Office, Data Systems Laboratory, coordinated the exchange of information and provided guidance and information. The analyses, results, and reporting were provided by Dr. R. Atkinson, Dr. D. Bond, Dr. B. Dasarathy, Mr. M. Lybanon, and Dr. H. Ramapriyan of Computer Sciences Corporation and Dr. R. Jayroe of the Information Sciences Division, Data Systems Laboratory.

2.1 DESCRIPTION OF DATA SETS

The data employed in this study were obtained from the computer compatible tapes of the four Landsat Multispectral Scanner (MSS) Images each consisting of information in four spectral bands:

1211-15493:	February 19, 1973
1265-15444:	April 14, 1973
1337-15490:	June 25, 1973
1607-15440:	March 22, 1974

The test site used as the object of the study is the 7-1/2 minute quadrangle known as the Bald Knob Quadrangle in the State of Tennessee. Located with its Southwest corner at latitude 35° 45'N, longitude 85° 30'W, the site is typical of much of the rural land in the southern Appalachian region, with rolling wooded hills and diversified agriculture practiced in the valleys. Community settlements are small and scattered.

A region containing the Bald Knob Quadrangle was extracted from each of the above images and the images were registered so that the four data sets on tape corresponded to exactly the same region. The image size was 200 lines with 260 pixels in each line. Due to the orientation of Landsat, this was approximately the minimum size of the rectangle containing the desired geographic region.

The April 14, 1973, image was processed by the classification algorithms, independently of the others, to produce maps of land cover. Unless otherwise specified, the classification maps in the sequel refer to this image. Also, classification maps were obtained by processing all four images simultaneously (using 16-dimensional feature vectors). These will be referred to as multitemporal classification maps.

Figure 2-1 shows the 16 images corresponding to the four bands in each of the four data sets mentioned above. They are arranged from left to right in the order of the Landsat multispectral scanner channels 4 (green: 0.5 to 0.6 μm), 5 (red: 0.6 to 0.7 μm), 6 (infrared: 0.7 to 0.8 μm), and 7 (infrared: 0.8 to 1.1 μm). The seasonal differences in the scene are striking, exemplified by the tonal differences, caused by variations in reflected sunlight, in the red band (second column from the left) for the spring, summer, and winter scenes. Unfortunately, the two infrared channels malfunctioned during the summer Landsat overpass, resulting in a high level of electrical noise (random fluctuations) in the data. The effect of this anomaly on the analysis could not be assessed, although it certainly prejudiced the quality of results obtained in

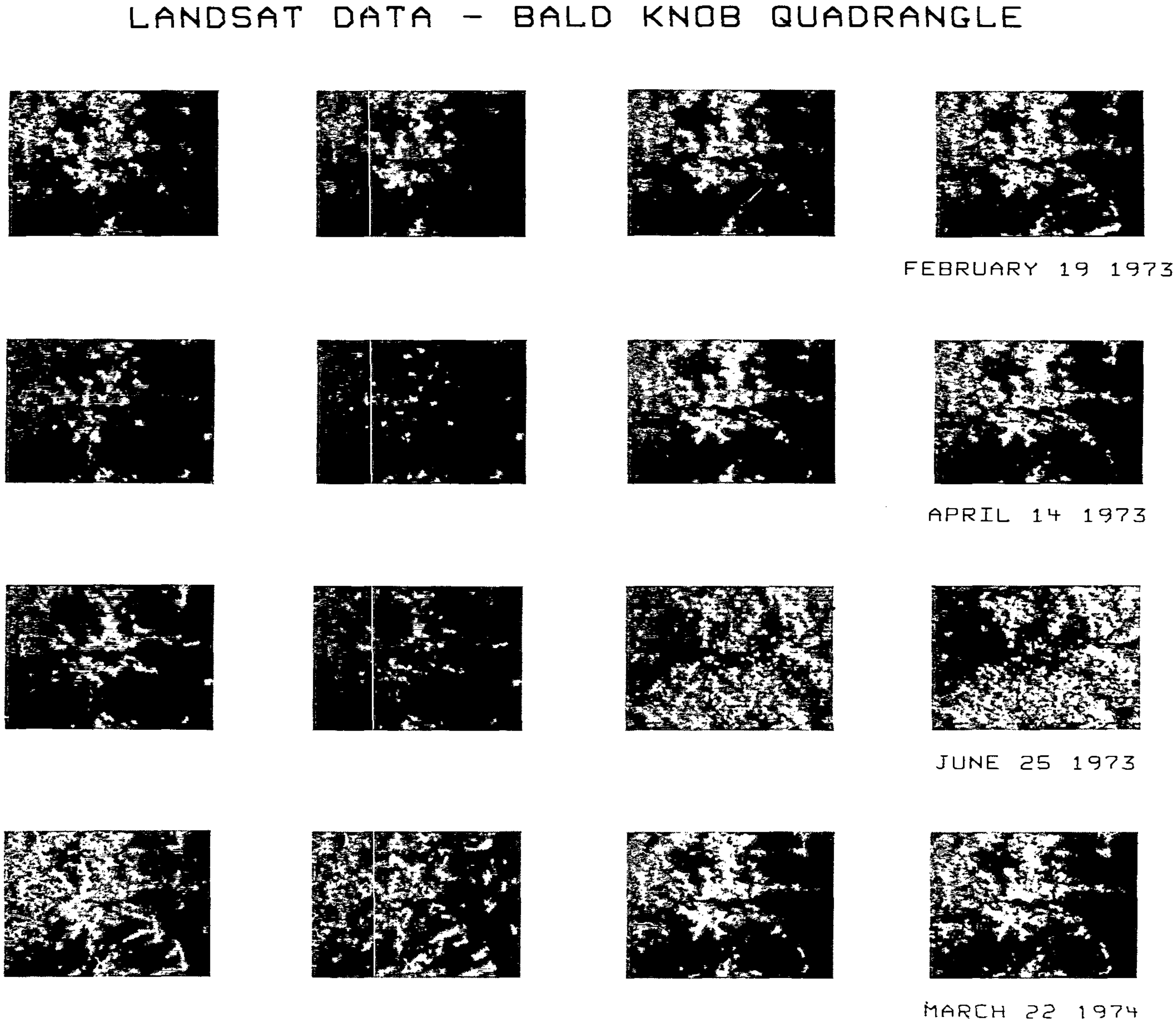


Figure 2-1. Landsat Images of the Bald Knob Test Site

multitemporal classification, since infrared reflectance in a summer scene is known to be a significant component in discriminating between land cover types.

Figure 2-2 shows one band of the April 14, 1973, image with the training samples identified by their boundary polygons (or by simply lines in the case of single-line sets of samples).

Also shown in Figure 2-2 is a preliminary classification map that illustrates the procedure, described in Section I-5.3, of improving the purity of training samples by slightly varying the location of training sites to assure, as far as possible, that they do not intersect class boundaries. The selection of training sites is a time consuming procedure, and usually requires several attempts before a satisfactory set is identified.

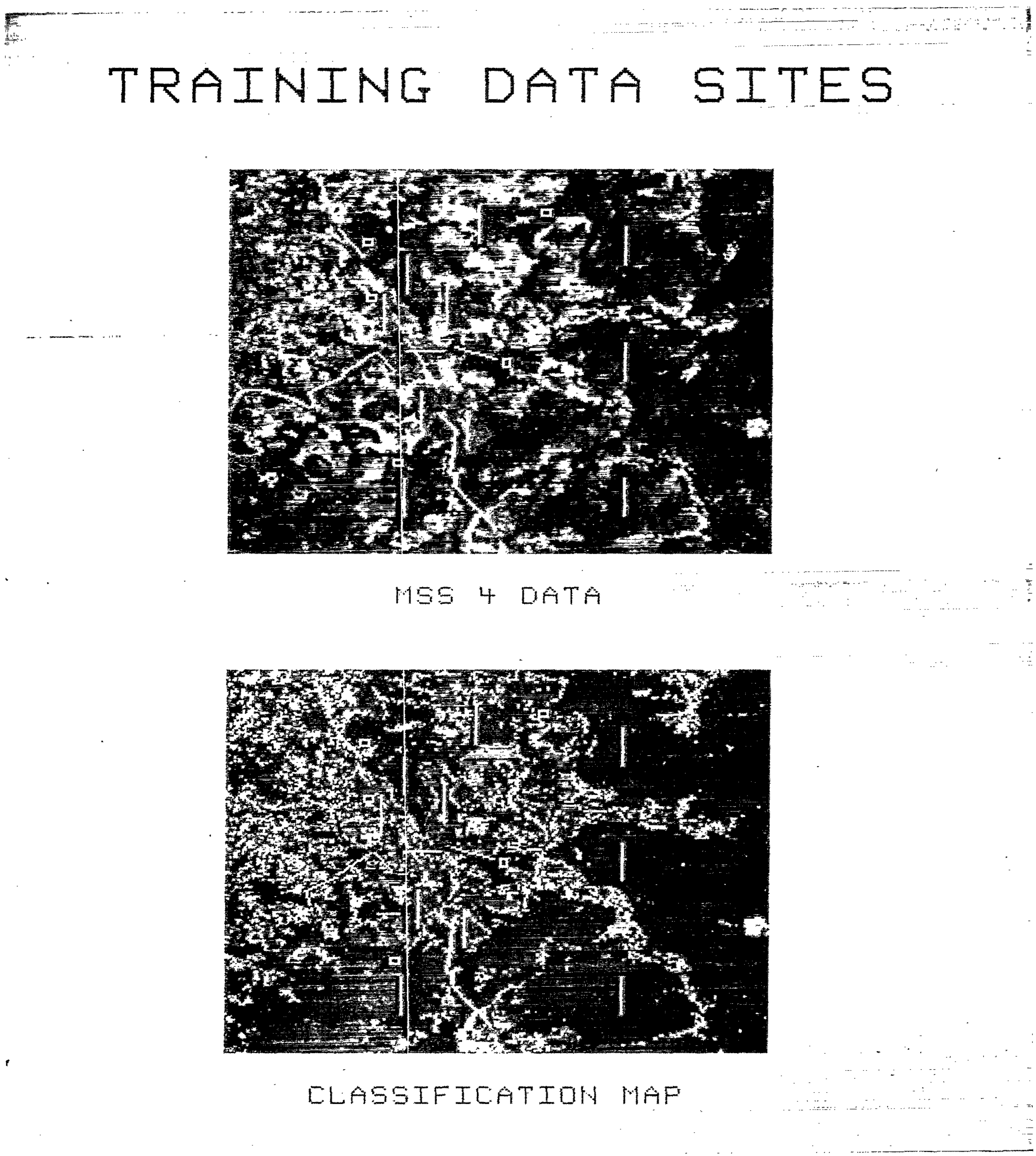


Figure 2-2. Training Site Locations in the Bald Knob Quadrangle

2.2 PREPARATION OF THE GROUND TRUTH MAP

A manually prepared ground truth map showing eight land use classes was supplied by the Tennessee State Planning Office. It was developed using information extracted by interpretation of orthophotoquads and hyperaltitude color infrared aerial photography. The classes were:

- Urban
- Transportation
- Agriculture
- Deciduous Forest
- Evergreen Forest
- Mixed Forest
- Water
- Strip Mining

The map was traced and copied in color using sufficiently distinct colors for each different class and omitting the annotations. This map was reduced to a 35 mm transparency and digitized using red, green, and blue filters. The resulting three-band image was then classified using a linear classification method, training samples being chosen from each of the (uniformly colored) areas. The output of this was compared with the original map and some minor manual editing was performed.

The scale and orientation of this map were different from those of the Landsat data. Since this map had to be compared with several maps produced from the Landsat data, it was more economical to distort it to the geometry of the Landsat data (rather than vice versa). The correction transformation was found using several control points on this map and a topographic map of the same region, employing a mean squared error minimization method* to find the transformation to UTM coordinates and using a previously determined transformation from UTM coordinates to Landsat pixel coordinates. The RMS error at the control points was approximately 0.5 Landsat pixel and the maximum error 1.2 pixels. The geometric correction was performed digitally, nearest neighbor values being used in resampling.

*"Geographic Referencing of Remotely Sensed Imagery Employing General Linear Transformations," M. Lybanon, Computer Sciences Corporation Memorandum Number 5E3030-1-4, Huntsville, Alabama, January 29, 1975.

The resulting map was used in all the comparisons. It will simply be referred to as the "Ground Truth Map" (GTM). This map is shown in Figure 2-3 and is color coded as follows:

Urban	-	Red	Evergreen Forest	-	Dark Green
Transportation	-	White	Mixed Forest	-	Green
Agriculture	-	Yellow	Water	-	Blue
Deciduous Forest	-	Pale Green	Strip Mining	-	Purple

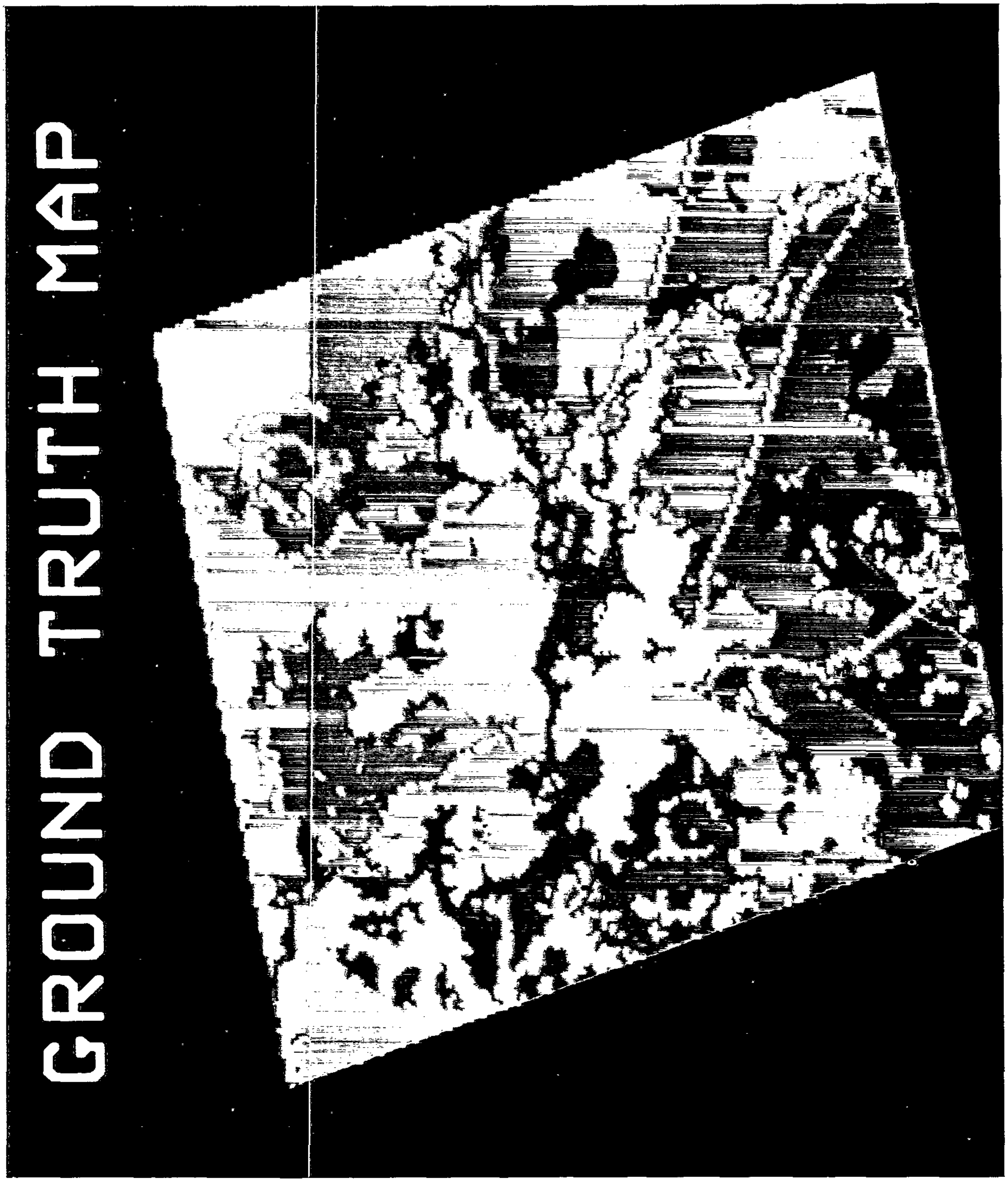


Figure 2-3. Ground Truth Map After Digitization and Distortion to Congruence with Landsat Scenes (for Color Code, refer to Text)

2.3 COMPARISONS WITH THE GROUND TRUTH MAP

2.3.1 Similarity Measures and the Bases for Comparison

The maps produced by each of the classification methods were compared with the GTM. Since the GTM was geometrically distorted to match the Landsat images, there was a likelihood of an error in registration of the order of one pixel. Therefore, in testing each of the supervised classification maps, nine joint histograms were obtained by considering the GTM and the eight possible shifts of it by one pixel. It was found that a shift to the left by one pixel yielded the greatest similarity measure in most cases, but the unshifted GTM gave almost the same result. Therefore, the unshifted GTM was taken as the correct reference and only the comparisons with it will be presented.

Note that the GTM has three forest classes--Deciduous, Evergreen, and Mixed. The mixed class is simply a mixture of the other two classes. Therefore, assignment of points in the mixed class in the GTM to Deciduous or Evergreen by a classifier should not be considered an error. This is reflected in the similarity measures defined between the supervised classifications and the GTM.

If A is the joint histogram between the GTM and a supervised classification map (see Section I-5.4), then the rows 4, 5, and 6 of A correspond respectively to Deciduous, Evergreen, and Mixed Forest classes. The classes in the supervised classification maps are Urban, Transportation, Agriculture, Deciduous Forest, Evergreen Forest, and Water. Therefore, the similarity measure is defined as

$$\frac{\text{Number of joint occurrences of similar classes}}{\text{Total number of points}}$$

$$= \frac{\sum_{i=1}^{6} a_{ii} - a_{66} + a_{76} + a_{64} + a_{65}}{\sum_{i=1}^{8} \sum_{j=1}^{6} a_{ij}}$$

However, in the case of the unsupervised maps, the correspondence of homogeneous clusters to the ground truth classes is not known beforehand and hence a simple expression such as the above cannot be derived. Therefore, a reassignment algorithm* is used to maximize the similarity measure, treating Mixed Forest as a separate class.

*"Constrained Assignment of Unsupervised Classification Numbers to Maximize Similarity with a Supervised Classification," H. K. Ramapriyan, Computer Sciences Corporation Memorandum Number 5E3090-1-5, Huntsville, AL, October 1975.

Similarity measures were also found by combining the forest classes. Then, in the case of the supervised maps the similarity measure is given by

$$\frac{\sum_{i=1}^{6} a_{ii} - a_{66} + a_{76} + a_{64} + a_{65} + a_{45} + a_{54}}{\sum_{i=1}^{8} \sum_{j=1}^{6} a_{ij}}$$

In the case of the unsupervised maps, the rows of A corresponding to the Forest classes are first added and then the reassignment of columns is made to maximize the similarity measure.

Further, the difference maps indicate that most errors occur along the boundaries between ground truth classes. This is partly attributable to: (i) possible errors in ground truth boundary determination, (ii) mixture of colors occurring along boundaries while digitizing the ground truth map, and (iii) errors in the determination of the geometric transformation. Therefore, it is appropriate to show difference images indicating locations of errors corresponding to boundary points in the GTM separately from those at interior points. In doing this, any point in the GTM is considered a boundary point if at least one of its four neighbors (left, right, top, and bottom) is different from it.

A third kind of similarity measure is determined by treating as erroneous classification only the interior dissimilarities between the GTM and the classification maps.

In subsections 2.3.2 through 2.3.8, the classification maps and similarity matrices for the algorithms described in Section II are presented. The normal similarity and the similarity measures derived through combining Forest classes and by suppressing boundary errors are quoted in each case. In a few instances, primarily for illustrative purposes, the dissimilarity map is also exhibited. A summary tabulation of the similarity measures is presented in subsection 2.3.9.

All of the supervised classification results were produced with only six classes, the "Mixed Forest" and "Strip Mining" classes being omitted, since the former is not different from the other two Forest classes, and the number of points in the latter (16 out of 52,000) is not statistically significant. In the similarity matrices that follow, the Ground Truth Map classes are ordered in rows, and the classification map classes are ordered in columns. For simplicity, the classes are numbered as follows:

	GTM		Classification Map
1	Urban	1	Urban
2	Transportation	2	Transportation
3	Agriculture	3	Agriculture
4	Deciduous Forest	4	Deciduous Forest
5	Evergreen Forest	5	Evergreen Forest
6	Mixed Forest	6	Water
7	Water		
8	Strip Mining		

The one exception to this convention applies to the Table Lookup Classifier ELLTAB, which produces a seventh "rejection" class due to the thresholding feature of the algorithm. In presenting the supervised classification maps, the color code was chosen to be approximately the same as that for the corresponding classes in the GTM.

The number of classes produced by the unsupervised classifiers varies depending on the technique and its manner of use. In this case the correspondences with the GTM were established manually by inspecting the joint histograms. The predominant classes were assigned colors corresponding to the respective classes in the GTM. The remaining classes were assigned arbitrary but distinct colors.

2.3.2 Density Slicing Classification Map (DSCM)

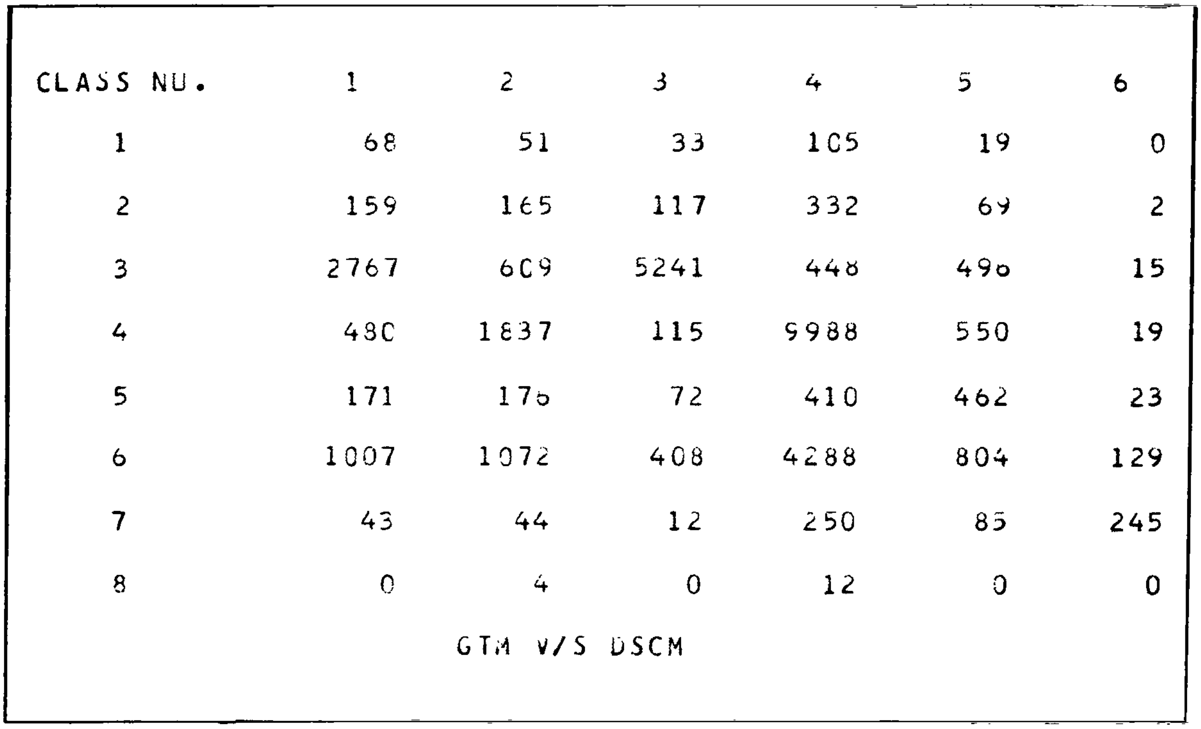
Application of the feature selector EFFECT to the April 1973 data set determined the optimum spectral bands and corresponding density ranges for separating the six classes to be those listed below.

Optimum Spectral Bands and Density Slices

Class	Spectral Band	Density Range
Water	4	0-12
Evergreen	2	0-21
Deciduous	3	0-36
Transportation	4	13-23
Agriculture	4	30-255
Urban	4	24-29

The resulting classification map appears in Figure 2-4, and the similarity matrix is given in Table 2-1.

Table 2-1. Similarity Matrix Between GTM and Density Slicing Classification Map



CLASS NU.	1	2	3	4	5	6
1	68	51	33	1C5	19	0
2	159	165	117	332	69	2
3	2767	6C9	5241	448	496	15
4	48C	1837	115	9988	550	19
5	171	176	72	410	462	23
6	1007	1072	408	4288	804	129
7	43	44	12	250	85	245
8	0	4	0	12	0	0

GTM V/S DSCM

The similarity matrix points out some of the key ambiguities in the classification. Observe for example, that the GTM contains only 276 Urban samples total, while the Density Slicing classifier identified 4695 Urban samples. Of these, 2767 were incorrectly assigned to the Agricultural class. This is not suprising in light of the density slices, which exhibit a broad spread of densities (30-255) associated with Agriculture, and a narrow spread of contiguous densities (24-29) in the same spectral band associated with Urban. This points up one of the weaknesses of separating classes on the basis of one spectral band only. Notice also that 250 of the GTM samples corresponding to Water were incorrectly assigned to the Deciduous Forest Class, again not suprising since the reflectance characteristics of Bands 3 and 4 are quite similar, and the Density range for Deciduous Forest in Band 3 (0-36) overlaps that for Water in Band 4 (0-12).

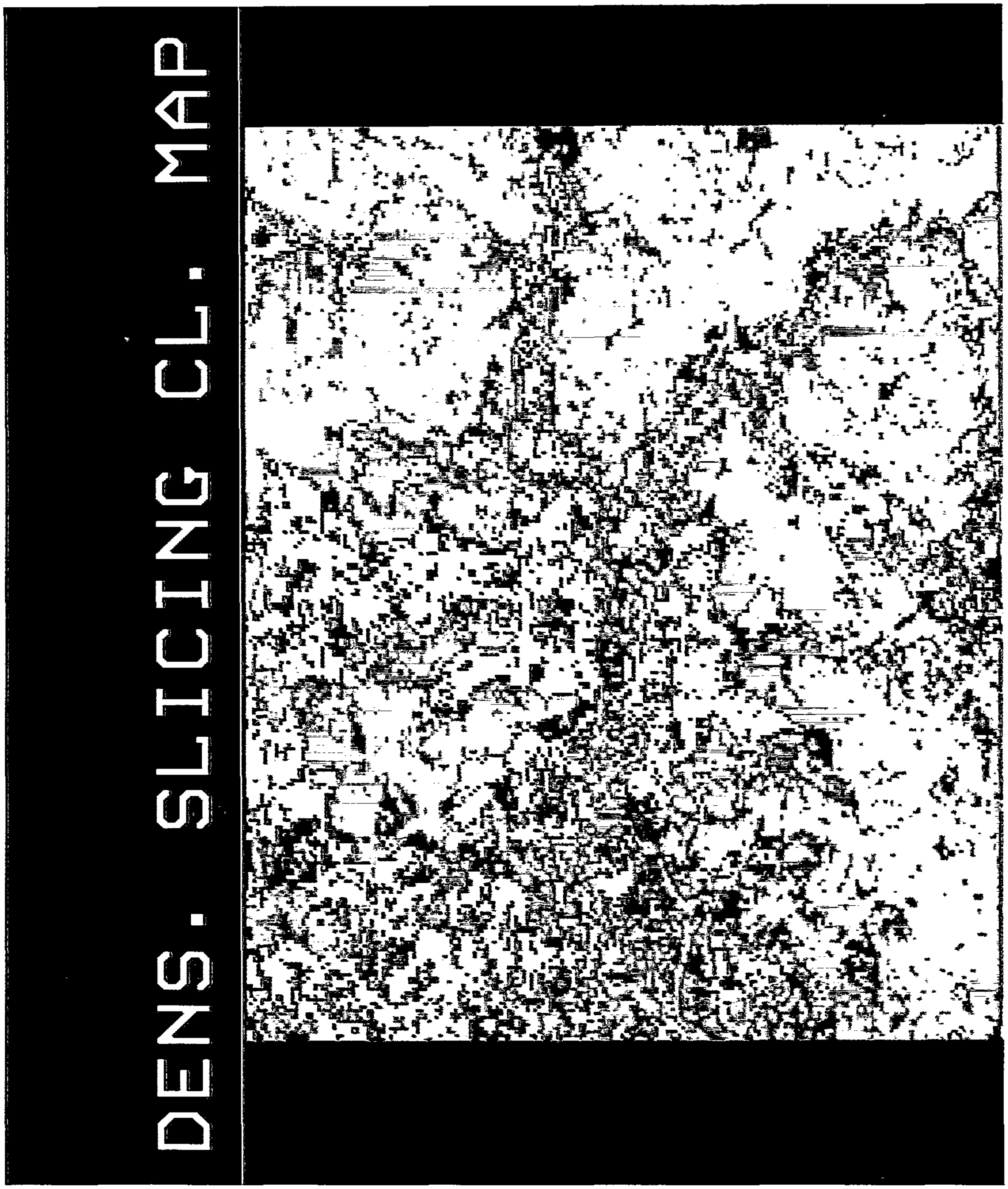


Figure 2-4. Density Slicing Classification Map

The similarity measures with the Ground Truth Map are

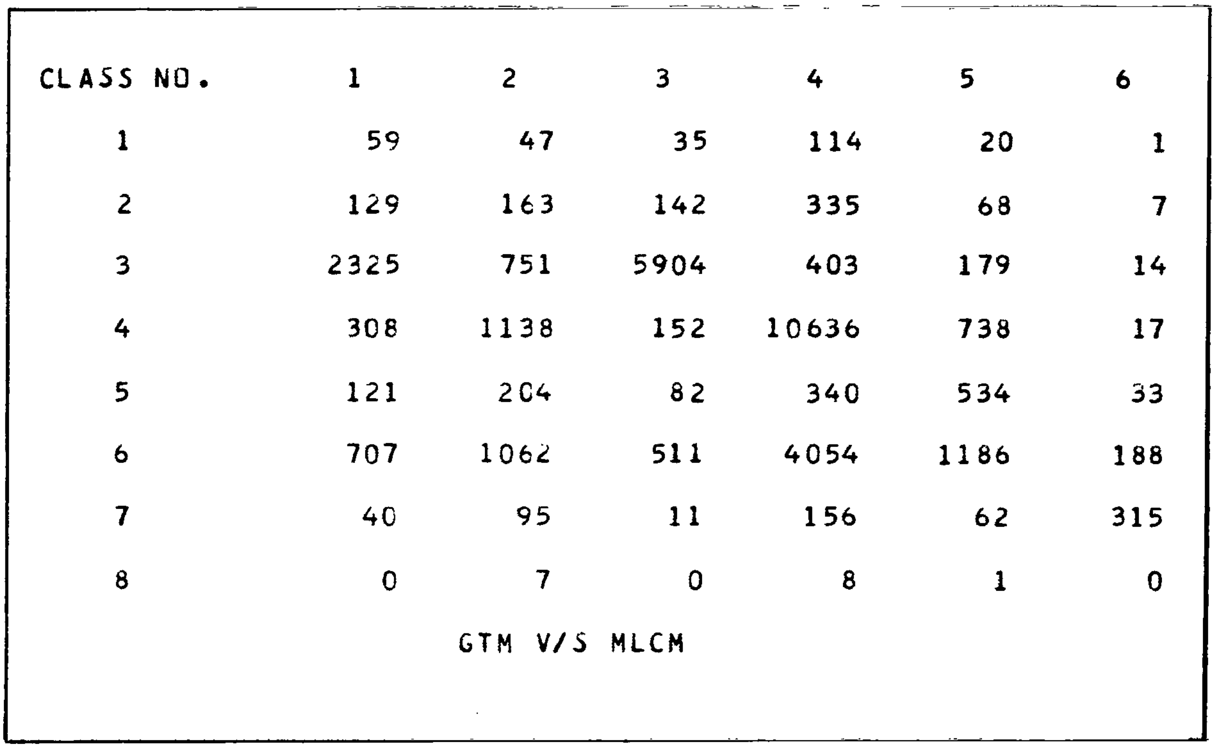
Normal	63.65 percent
Forest Merged	66.53 percent
Boundary Errors Ignored	84.89 percent

2.3.3 Maximum Likelihood Classification Maps

The April 1973 Data

The Maximum Likelihood Classification Map (MLCM) appears in Figure 2-5 and the corresponding dissimilarity map in Figure 2-6. Table 2-2 shows the Similarity matrix, which indicates similar trends in false assignments as applied to the density slicing classifier, though not quite as extreme. Again the confusion between Urban and Agriculture is apparent, though the confusion between Water and Deciduous Forest is not quite so serious, no doubt due to multiple band correlation implicit in the Maximum Likelihood scheme. As seen from Figure 2-6, the majority of the points of dissimilarity (colored grey) lie along boundaries.

Table 2-2. Similarity Matrix Between GTM and Maximum Likelihood Classification Map



CLASS NO.	1	2	3	4	5	6
1	59	47	35	114	20	1
2	129	163	142	335	68	7
3	2325	751	5904	403	179	14
4	308	1138	152	10636	738	17
5	121	204	82	340	534	33
6	707	1062	511	4054	1186	188
7	40	95	11	156	62	315
8	0	7	0	8	1	0

GTM V/S MLCM

The three measures of similarity with the Ground Truth Map are

Normal	68.41 percent
Forest Merged	71.65 percent
Boundary Errors Ignored	88.86 percent

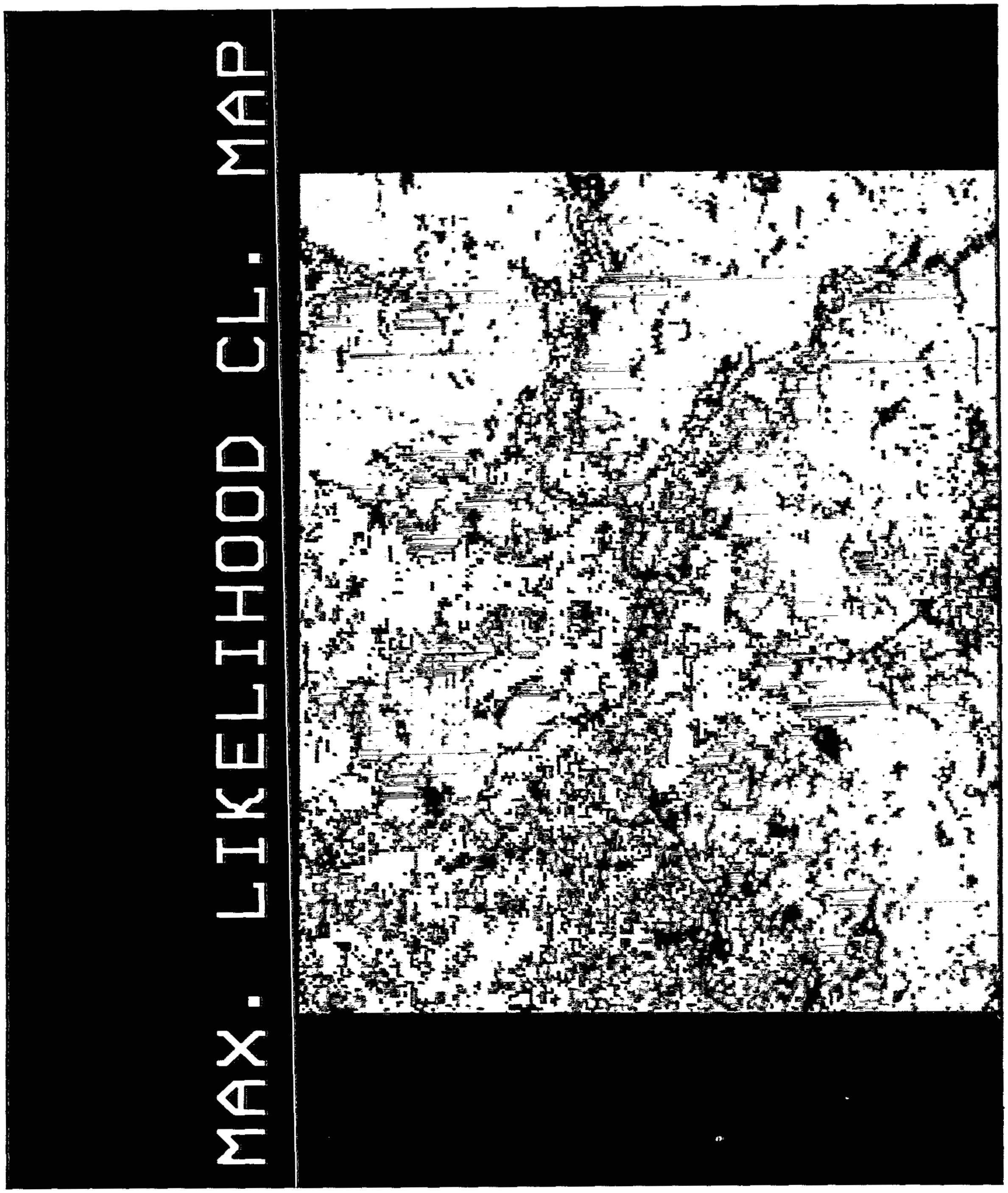


Figure 2-5. Maximum Likelihood Classification Map

Multitemporal Data

All 16 bands of the data exhibited in Figure 2-1 were also classified by the Maximum Likelihood Classifier, and this result appears in Figure 2-7 as the Multitemporal Maximum Likelihood Classification Map (MLMCM). Figure 2-8 exhibits the corresponding dissimilarity map and the similarity matrix is given in Table 2-3.

Table 2-3. Similarity Matrix Between GTM and Multitemporal Maximum Likelihood Classification Map

CLASS NO.	1	2	3	4	5	6
1	58	49	44	103	21	1
2	101	247	175	240	76	5
3	1369	1126	6750	203	96	32
4	345	891	210	10611	899	33
5	92	304	131	233	498	56
6	589	1523	747	3485	1093	271
7	36	133	34	69	48	359
8	0	9	0	7	0	0

GTM V/S MLMCM

For this particular data set, the reduction in confusion by incorporating this data from other seasons is not as striking as would be hoped.

For example, in residential urban areas, the summer tree canopy causes understandable confusion with the Forest class, but use of winter data enables this anomaly to be controlled. Here, however, 1136 of the classifier Urban samples were actually Forest in the April data (Table 2-2), but addition of other season data reduced this misclassification only to 1026 samples. The Transportation class showed a slight improvement with multitemporal data. For this class, the GTM indicated a total of 844 Transportation samples, but the April classification identified only 163 as Transportation, while 335 were identified as Deciduous Forest. With multitemporal data, these assignments changed to 247 as Transportation and 240 as Deciduous Forest.

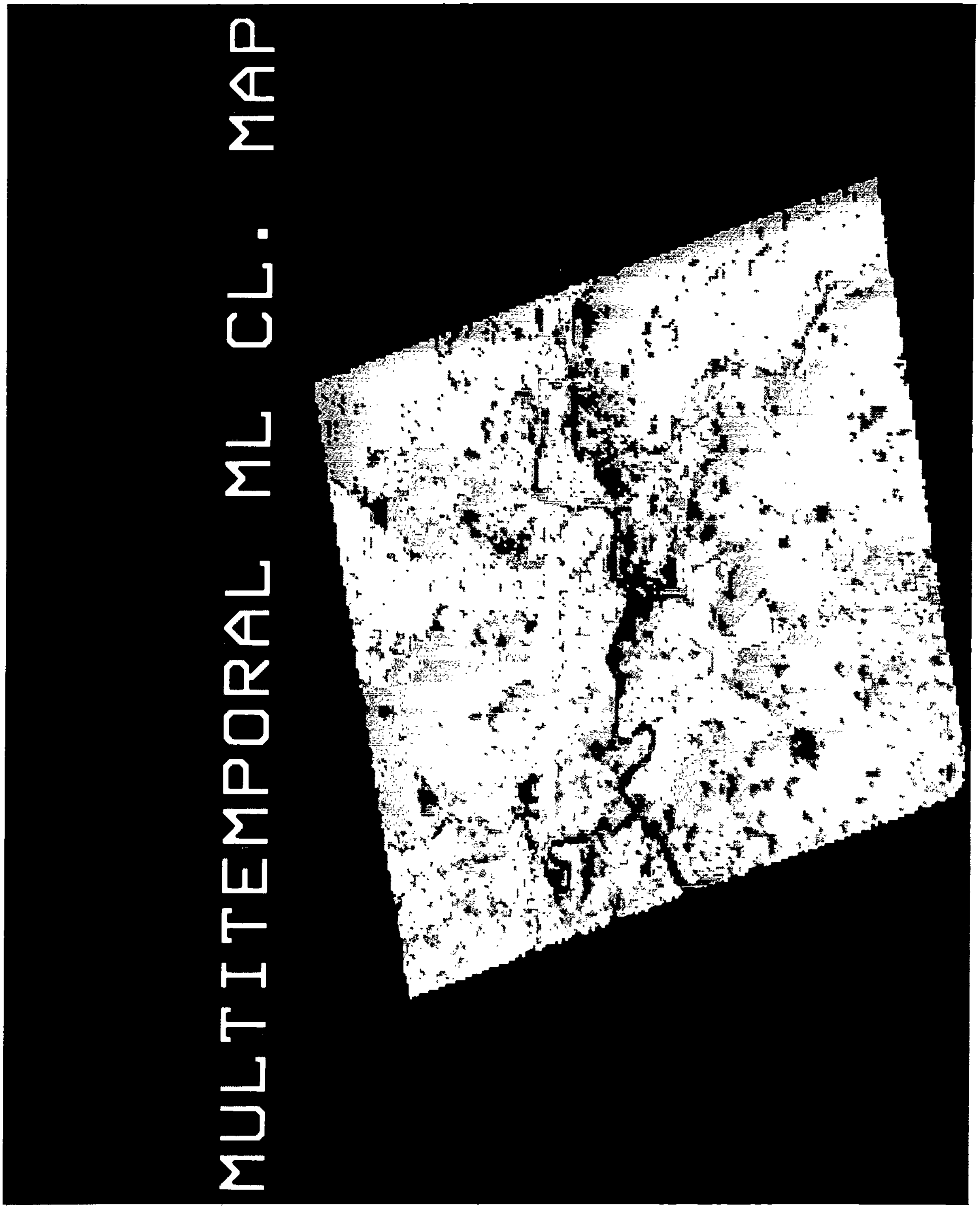


Figure 2-7. Multitemporal Maximum Likelihood Classification Map

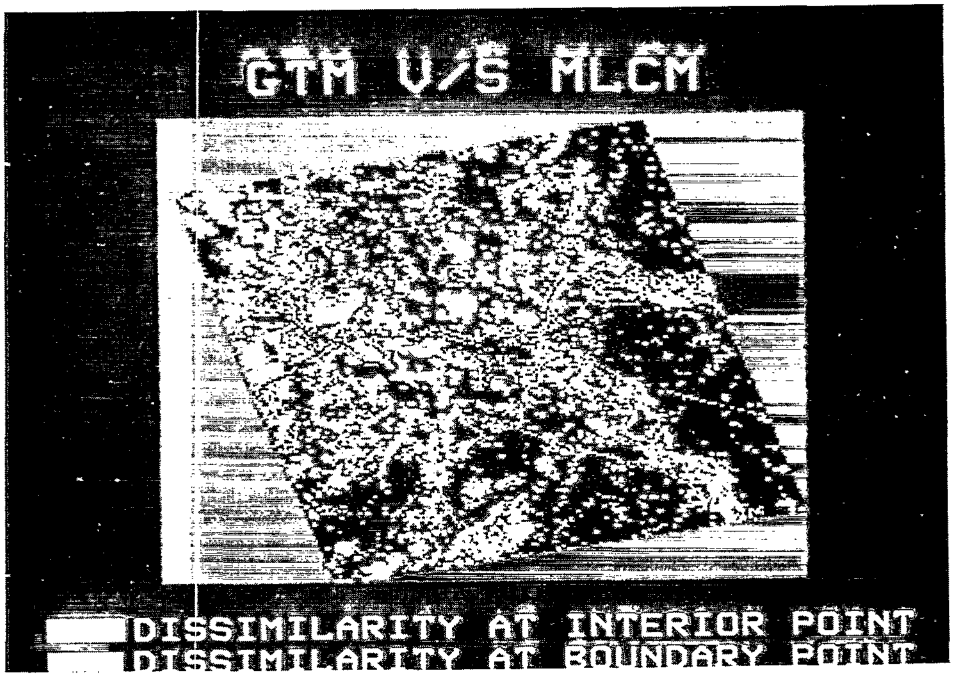

GTM V/S MLCM
DISSIMILARITY AT INTERIOR POINT
DISSIMILARITY AT BOUNDARY POINT

Figure 2-6. Dissimilarity Map Between GTM and Maximum Likelihood

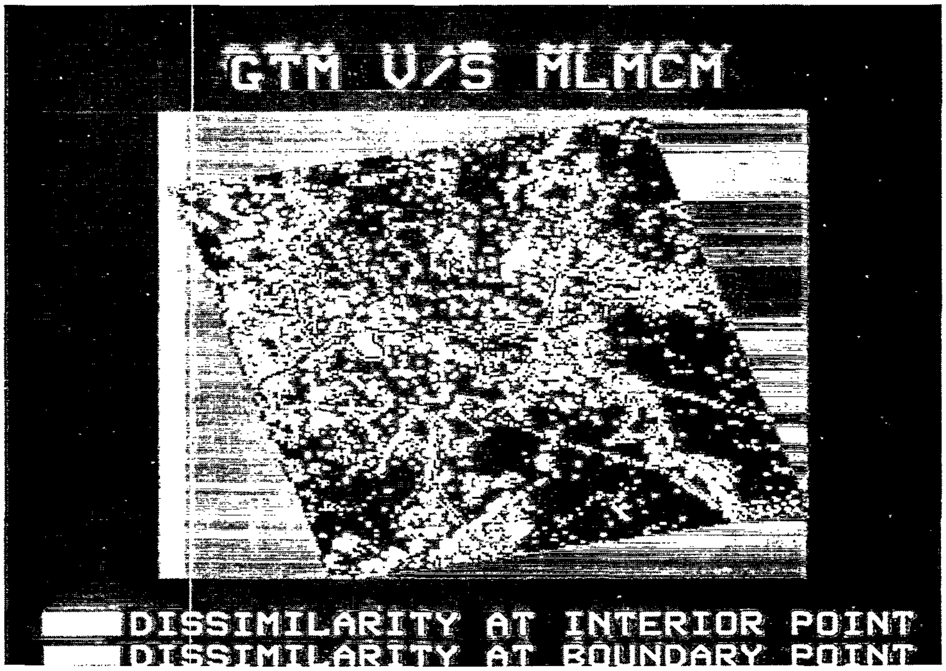

GTM V/S MLMCM
DISSIMILARITY AT INTERIOR POINT
DISSIMILARITY AT BOUNDARY POINT

Figure 2-8. Dissimilarity Map Between GTM and Multitemporal Maximum Likelihood

The most noticeable improvement appeared in reducing confusion at the Urban/ Agriculture interface. Of the 9576 Agriculture samples in the GTM, the April classification identified 5904 (61.7 percent) as Agriculture and 2325 (24.2 percent) as Urban. With multitemporal data these assignments changed to 6750 (70.5 percent) as Agriculture and 1369 (14.2 percent) as Urban.

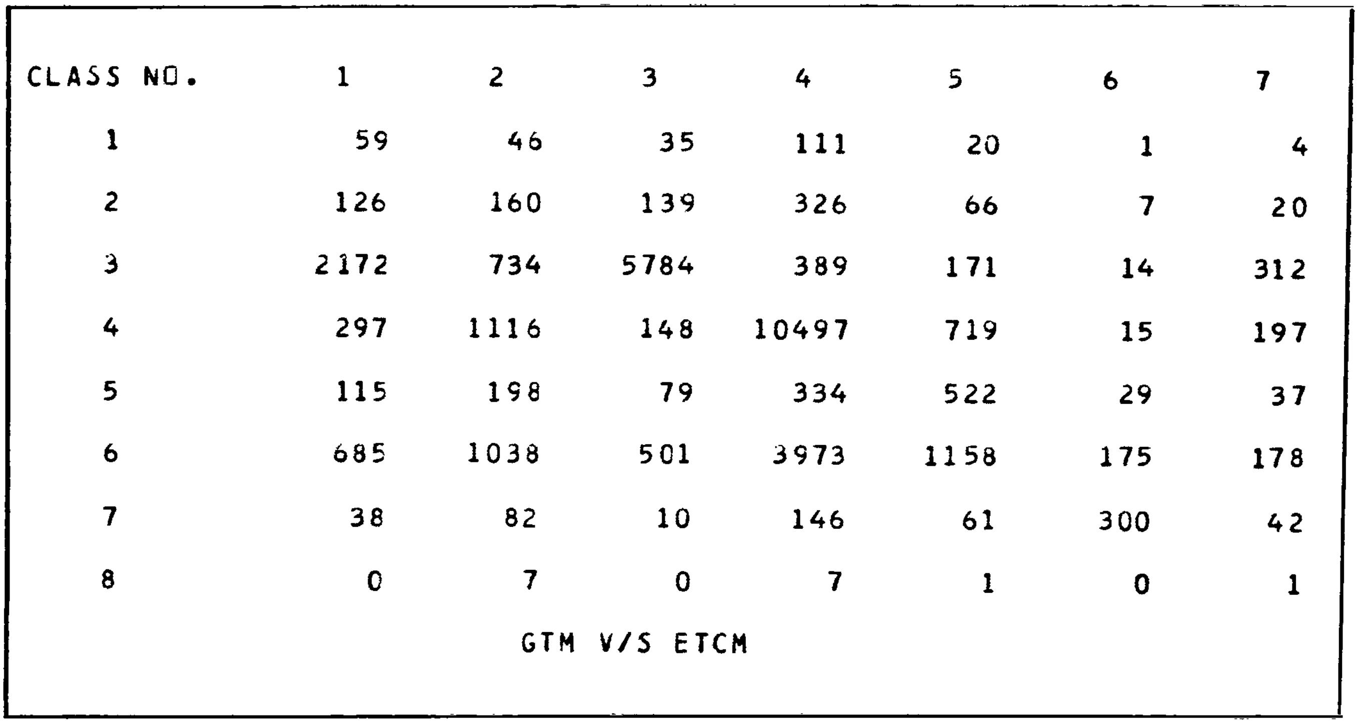
The three measures of similarity between the Ground Truth Map and the Multitemporal Maximum Likelihood Classification are

Normal	69.16 percent
Forest Merged	72.55 percent
Boundary Errors Ignored	90.31 percent

2.3.4 Table Lookup (ELLTAB)

Since ELLTAB employs a maximum likelihood classification table, its result should not differ appreciably from that of the maximum likelihood classifier, and this expectation is confirmed by comparing the ELLTAB Classification Map (ETCM), Figure 2-9 with that in Figure 2-5, and the similarity matrix Table 2-4 with that in Table 2-2.

Table 2-4. Similarity Matrix Between GTM and ELLTAB Classification Map



CLASS NO.	1	2	3	4	5	6	7
1	59	46	35	111	20	1	4
2	126	160	139	326	66	7	20
3	2172	734	5784	389	171	14	312
4	297	1116	148	10497	719	15	197
5	115	198	79	334	522	29	37
6	685	1038	501	3973	1158	175	178
7	38	82	10	146	61	300	42
8	0	7	0	7	1	0	1

GTM V/S ETCM

The table entries differ only slightly, the majority of the 791 samples rejected by the thresholding criterion (column 7) being in the Agriculture and Forest classes. These cause a slight reduction in similarity measures, which for this case are

Normal	67.22 percent
Forest Merged	70.25 percent
Boundary Errors Ignored	88.06 percent

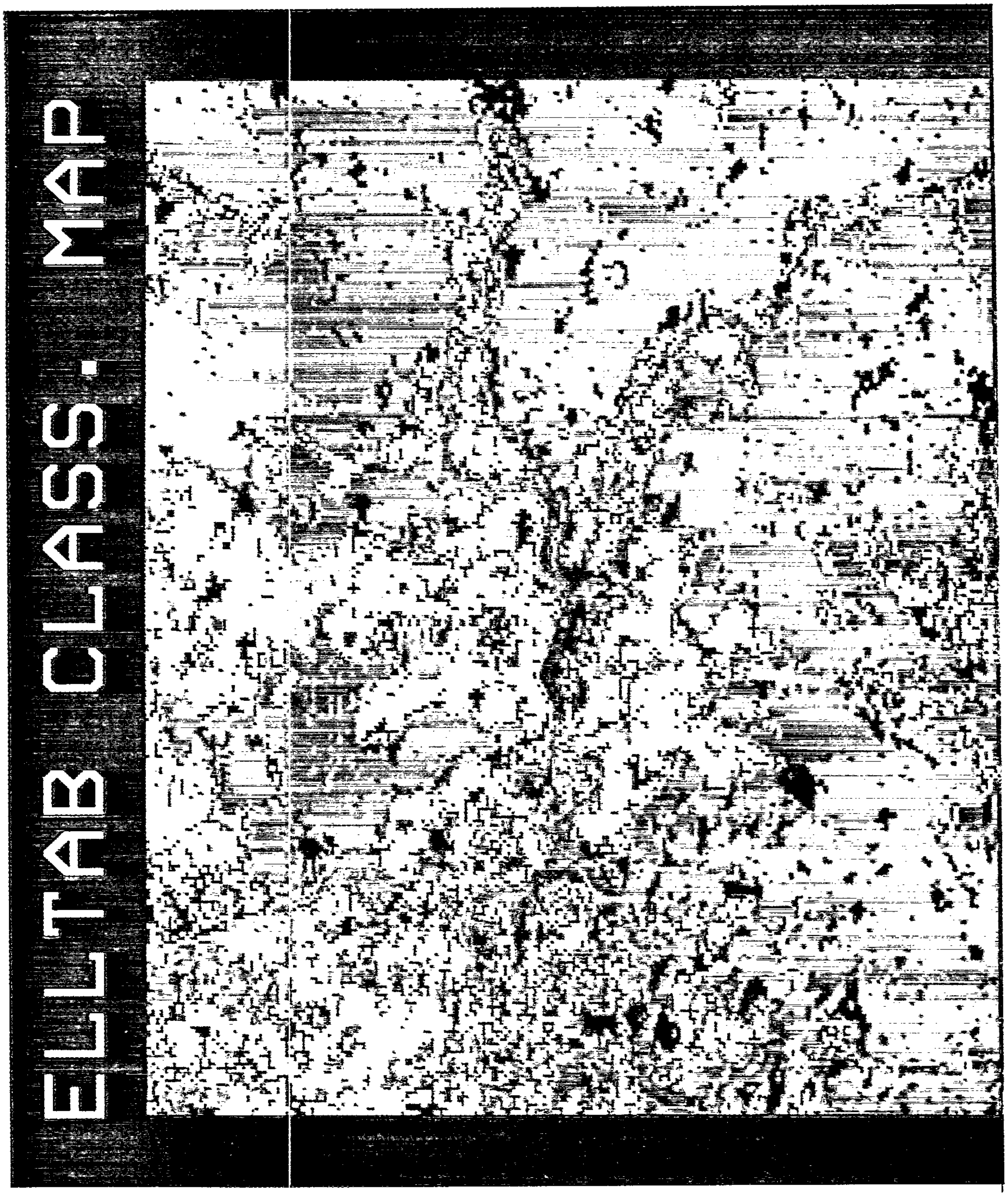


Figure 2-9. Table Lookup Classification Map

2.3.5 Linear Sequential

The April 1973 Data

The Linear Classification Map (LCM) appears in Figure 2-10, and the corresponding similarity matrix in Table 2-5.

Table 2-5. Similarity Matrix Between GTM and Linear Sequential Classification Map

CLASS NO.	1	2	3	4	5	6
1	50	36	41	121	28	0
2	101	151	164	355	72	1
3	1811	610	6308	609	233	5
4	263	1229	188	10308	999	2
5	85	133	107	477	486	26
6	599	818	602	4506	1033	150
7	28	62	17	156	126	290
8	0	7	0	9	0	0

GTM V/S LCM

These results differ only slightly from the single season Maximum Likelihood Classification and areas of confusion are similar. While there is less confusion between Urban and Agriculture, there is more confusion between Agriculture and Deciduous and Evergreen Forest. Similar observations may be made across the matrix.

The three measures of similarity with the ground truth map for this case are

Normal	69.25 percent
Forest Merged	73.35 percent
Boundary Errors Ignored	88.66 percent

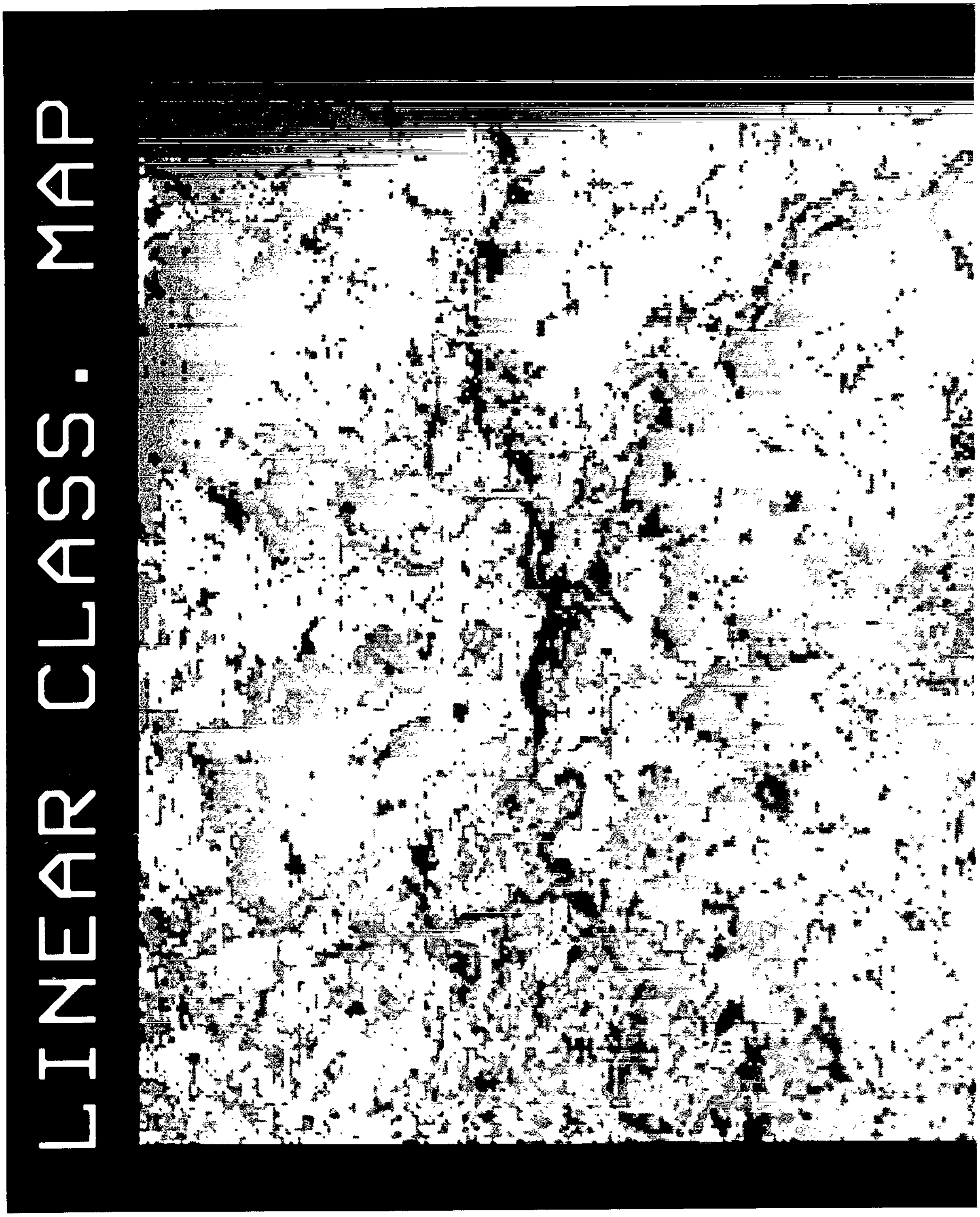


Figure 2-10. Linear Sequential Classification Map

Multitemporal Data

All 16 bands of the data exhibited in Figure 2-1 were also classified by the Linear Sequential Classifier, and this result appears in Figure 2-11 as the Linear Multitemporal Classification Map (LMCM). The corresponding similarity matrix is given in Table 2-6.

Table 2-6. Similarity Matrix Between GTM and Multitemporal Linear Classification Map

CLASS NO.	1	2	3	4	5	6
1	49	35	43	126	22	1
2	89	149	160	363	80	3
3	1714	839	6357	416	234	16
4	240	409	223	11309	803	5
5	83	150	125	454	470	32
6	552	708	696	4547	1026	179
7	34	72	21	113	153	286
8	0	7	0	8	1	0

GTM V/S LMCM

Here the most striking improvement resulting from multitemporal data shows up in reducing confusion between Deciduous Forest and Transporation. Of the 12989 Deciduous samples in the GTM, 10308 (79 percent) were classified as Deciduous Forest and 1229 (9 percent) as Transportation, using single season data. With multitemporal data, these assignments changed to 11309 (87 percent) Deciduous Forest and 409 (3 percent) Transportation. Although the similarity matrices show only slight variations, it is worth observing that the Transportation routes apparent in the lower right side of the GTM and Maximum Likelihood Classifications, do not appear in Figures 2-10 and 2-11, and these in fact show up as boundary dissimilarities in the Linear dissimilarity maps. This suggests that the linear classifier may be slightly more sensitive to boundary effects (for example malregistration) than the maximum likelihood classifiers, a conjecture supported by the similarity measures below.

Normal	72.43 percent
Forest Merged	76.19 percent
Boundary Errors Ignored	91.37 percent

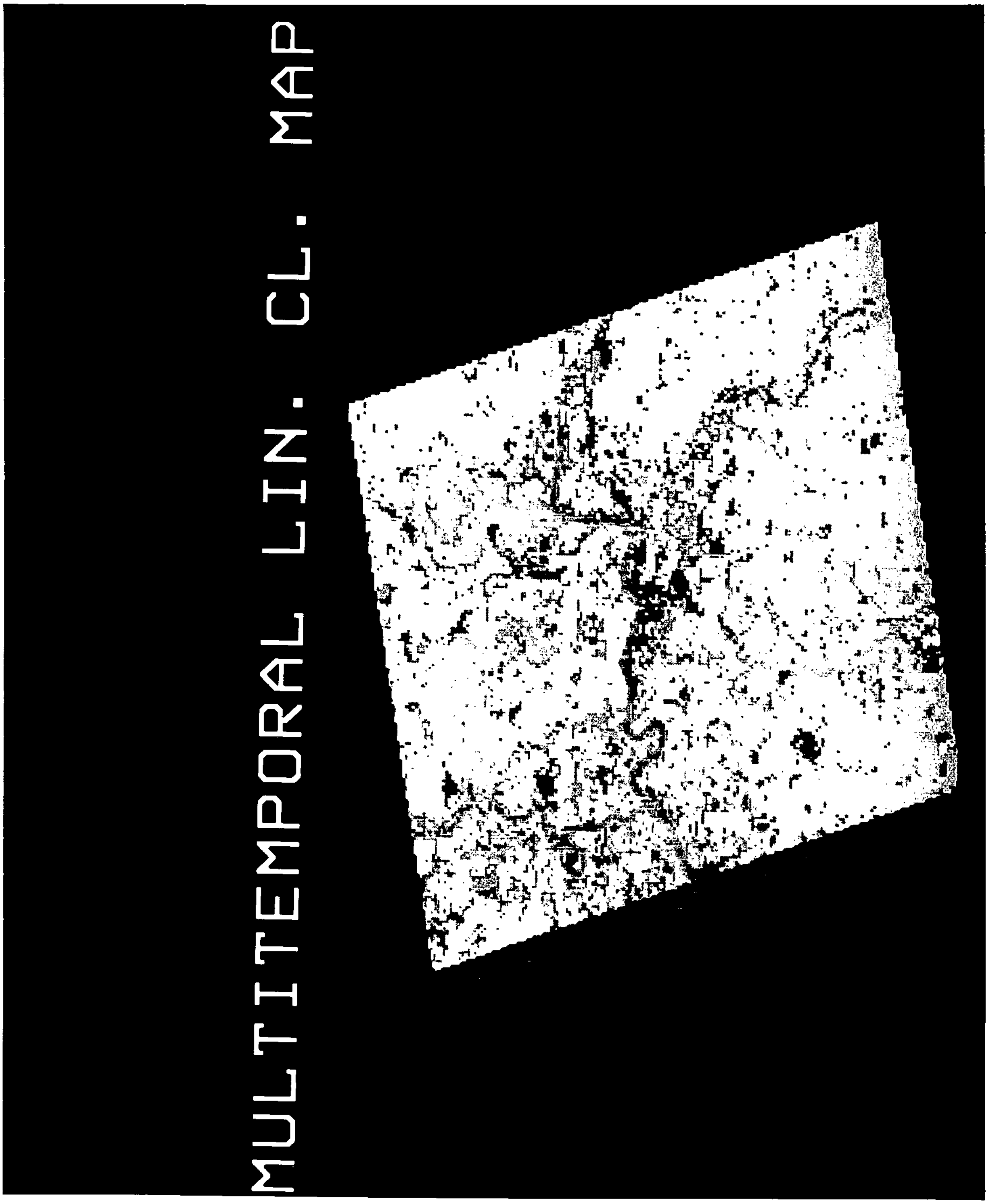


Figure 2-11. Multitemporal Linear Sequential Classification Map

2.3.6 Binary Classifier

The Binary Classifier is the most rudimentary of the unsupervised classification methods assessed, based as it is sole on comparisons of the magnitudes of feature vector components. The Binary Cluster Map (BCM) appears as Figure 2-12, and the associated similarity matrix is given in Table 2-7.

The Binary Classifier identified 11 distinct clusters in the data, and it is apparent from Table 2-7 that the most populous clusters are 1, 2, 3, 4, and 6. Because of the distributions of these populations, there is no clear correspondence between any one cluster and a single ground truth class. While one may be tempted to associate cluster 4 with the Agriculture class, and cluster 1 with the Deciduous Forest class, these associations are far from unique, since clusters 1 and 3 contribute significantly also to the Agriculture class while cluster 1 is the major contributor to the Mixed Forest Class. Clearly spectral resolution in the original data is too low to result in well defined homogenous clusters of reflectance values from vegetative land cover.

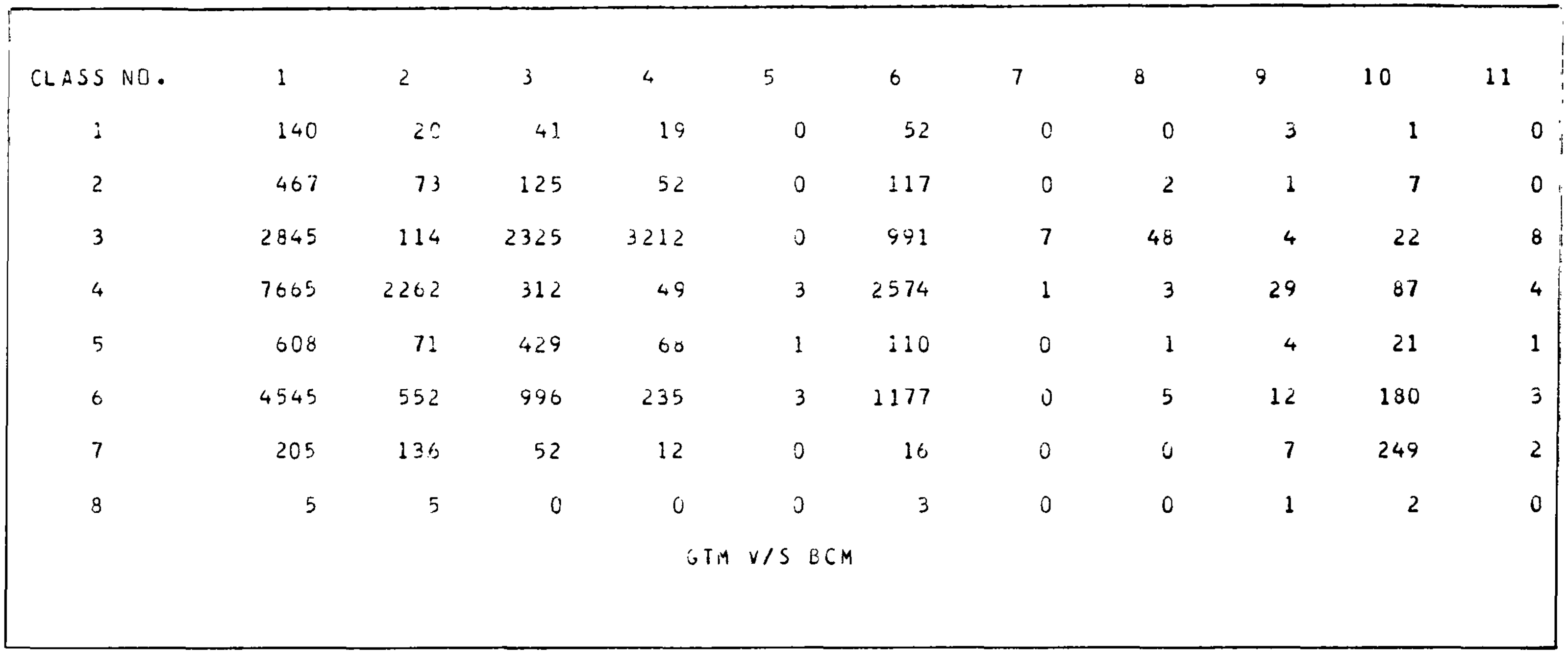
As described in Section III-2.3.1, the basis used for calculating similarity with the GTM is to reassign clusters, effectively merging clusters, in order to maximize the similarity measure subject to the constraint that the total number of remaining clusters equals the number of known ground truth classes.

The three measures of similarity with the GTM are

Normal	54.79 percent
Forests Merged	76.18 percent
Boundary Errors Ignored	90.05 percent

Also, if the GTM "Mixed Forest" samples are treated as either Deciduous Forest or Evergreen Forest, the normal similarity measure increases to 73.59 percent.

Table 2-7. Similarity Matrix Between GTM and Binary Cluster Map



CLASS NO.	1	2	3	4	5	6	7	8	9	10	11
1	140	20	41	19	0	52	0	0	3	1	0
2	467	73	125	52	0	117	0	2	1	7	0
3	2845	114	2325	3212	0	991	7	48	4	22	8
4	7665	2262	312	49	3	2574	1	3	29	87	4
5	608	71	429	68	1	110	0	1	4	21	1
6	4545	552	996	235	3	1177	0	5	12	180	3
7	205	136	52	12	0	16	0	0	7	249	2
8	5	5	0	0	0	3	0	0	1	2	0

GTM V/S BCM

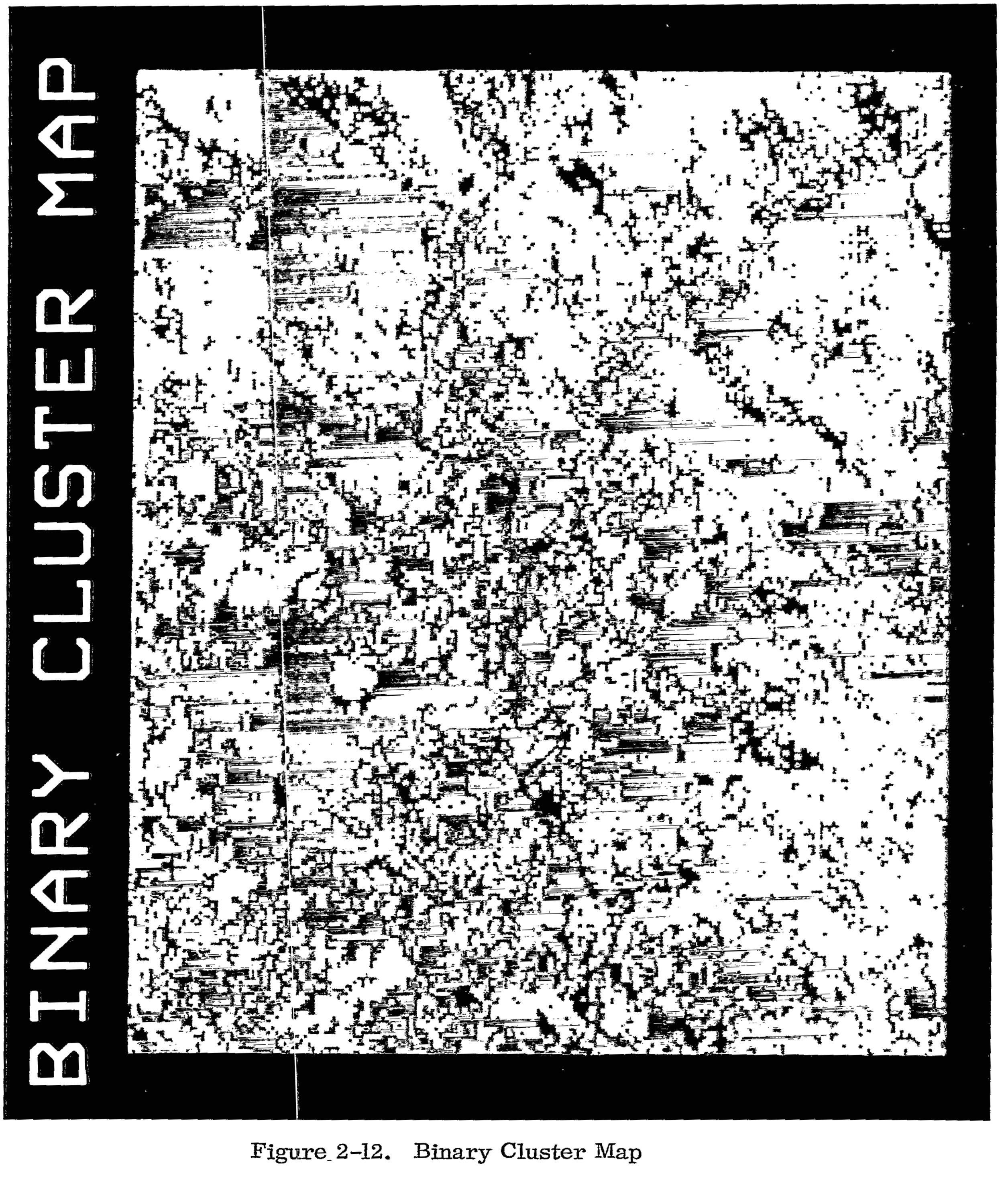


Figure 2-12. Binary Cluster Map

2.3.7 Spatial and Spectral Clustering Program (SSCP)

The Spatial and Spectral Clustering Program identified six homogenous clusters. As seen from the similarity matrix Table 2-8, there are few low-population clusters comparable to the low-population GTM classes (Urban, Water, Strip-mining). This is to be expected because of the spatial "windowing" property of the algorithm, which results in weak discrimination of spatially small features. The SSCP classification map appears in Figure 2-13.

Table 2-8. Similarity Matrix Between GTM and SSCP Classification Map

CLASS NO.	1	2	3	4	5	6	7
1	0	51	143	3	69	3	7
2	0	172	468	23	153	6	22
3	0	1913	1127	1437	4451	624	24
4	0	799	11003	20	203	4	960
5	0	517	649	17	112	3	16
6	0	1408	5318	63	719	26	174
7	0	412	209	1	30	1	26
8	0	6	9	0	0	0	1

GTM V/S SSCPCM

The three similarity measures for this case are

Normal	50.64 percent
Forests Merged	65.45 percent
Boundary Errors Ignored	82.24 percent

Also treating as correct the assignments of GTM "Mixed Forest" samples to either Deciduous Forest or Evergreen Forest, the normal similarity increased to 66.64 percent.

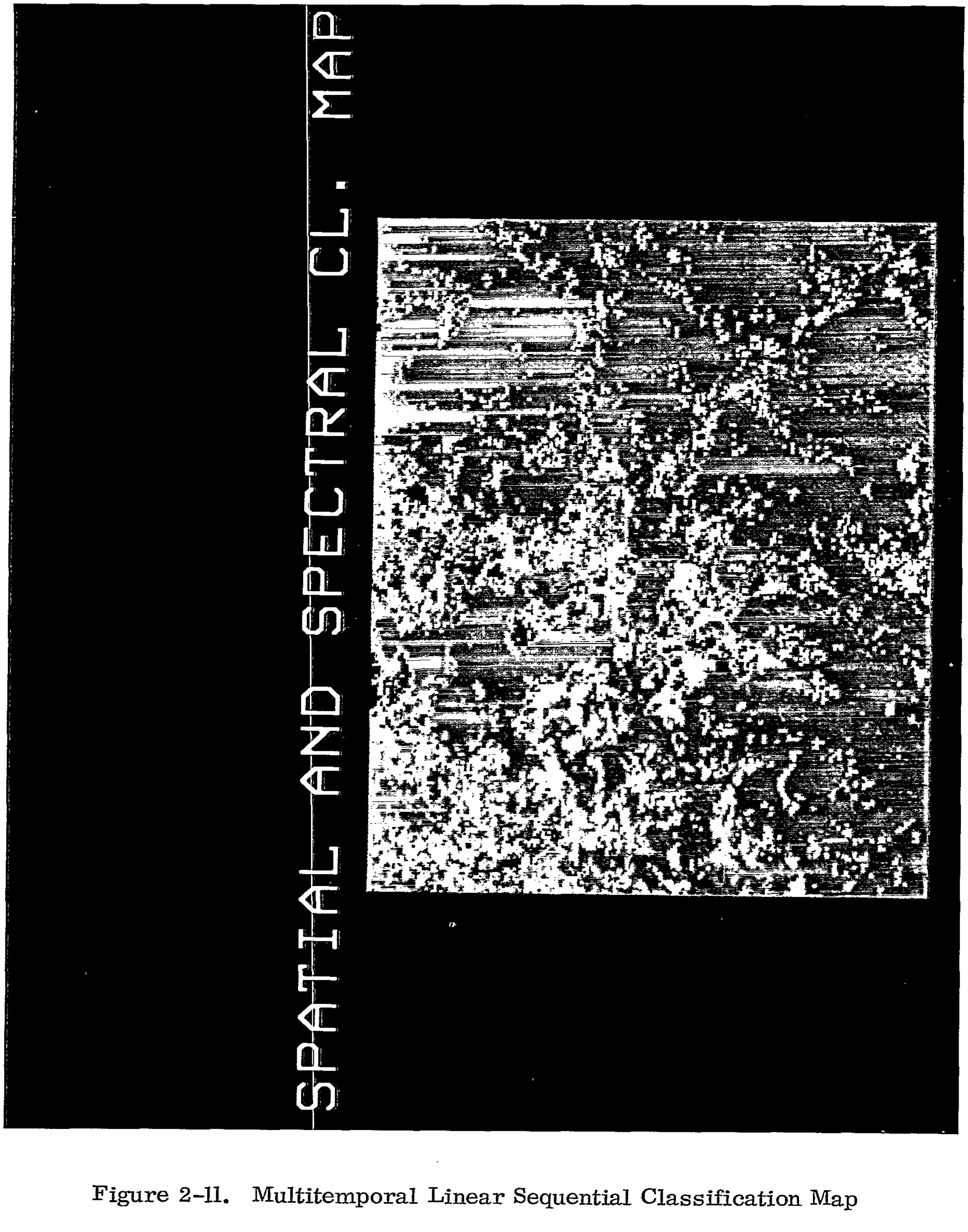


Figure 2-11. Multitemporal Linear Sequential Classification Map

2.3.8 HINDU Classifier

The Histogram dependent clustering technique identified nine homogenous clusters in the data. The similarity matrix Table 2-9 shows that, as with the other cluster techniques, there is no unambiguous association between individual clusters and GTM classes. The HINDU classification map is shown in Figure 2-14.

Following reassignment to maximize similarity, subject to the constraint of preserving eight clusters, the similarity measures between GTM and the HINDU classification map become

Normal	52.18 percent
Forest Merged	82.73 percent
Boundary Errors Ignored	95.14 percent

Also treating as correct the assignments of GTM "Mixed Forest" samples to either Deciduous Forest or Evergreen Forest, the normal similarity increased to 64.43 percent.

Table 2-9. Similarity Matrix Between GTM and HINDU Classification Map

CLASS NO.	1	2	3	4	5	6	7	8	9
1	95	105	46	4	11	10	5	0	0
2	301	324	122	31	31	25	8	2	0
3	404	1760	4024	1935	914	246	271	21	1
4	8579	4007	156	31	54	73	4	77	8
5	725	424	97	12	16	11	1	17	11
6	4037	2690	540	71	106	77	20	112	55
7	257	114	22	3	1	4	0	148	130
8	10	2	0	0	0	4	0	0	0

GTM V/S HINDUCM

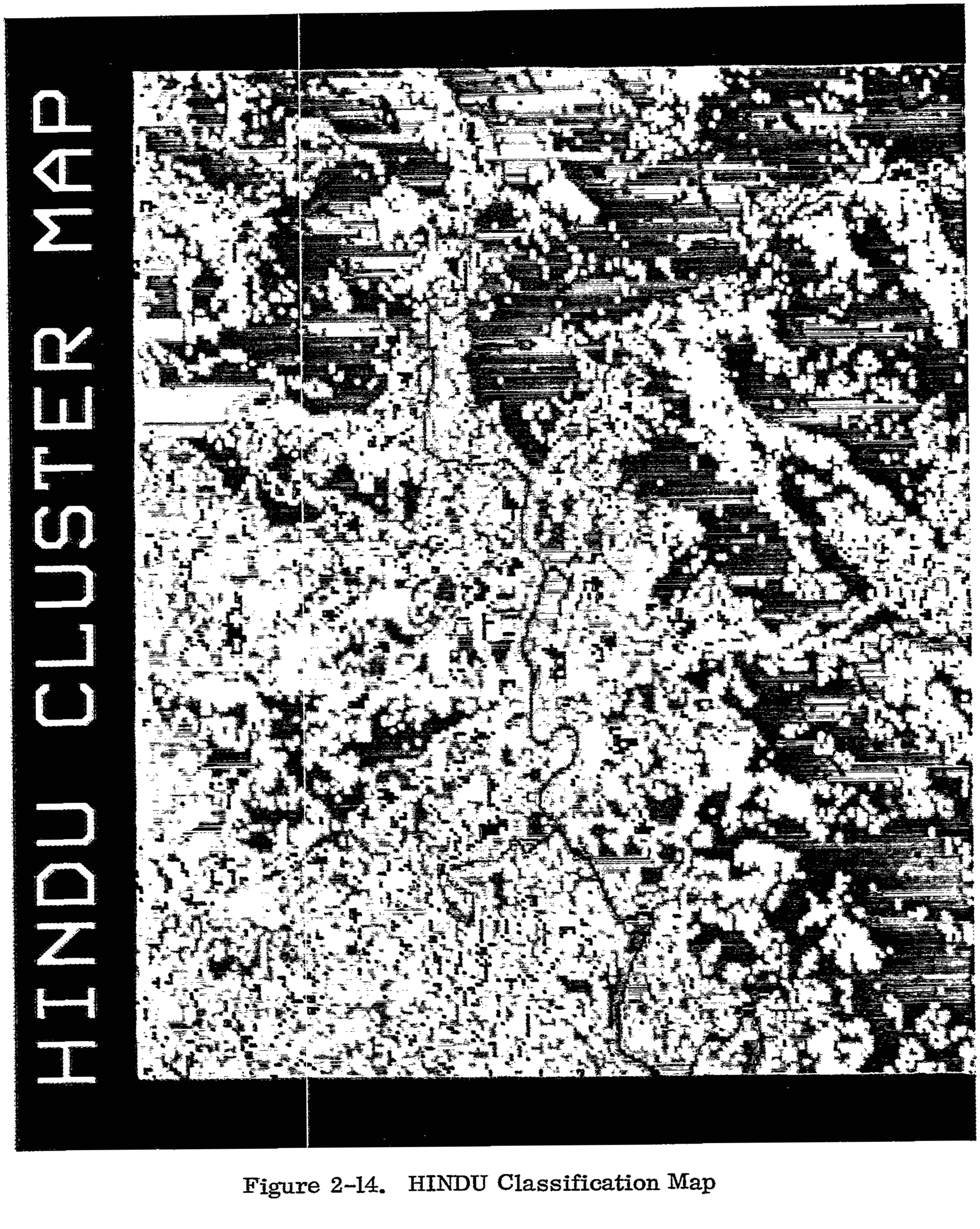


Figure 2-14. HINDU Classification Map

2.3.9 Comparisons with the Ground Truth Map — Summary of Results

For convenience of comparison between methods, the similarity measures between the Ground Truth Map and the Supervised Classification Maps are collected in Table 2-10. The similarity measures between the Ground Truth Map and the Unsupervised Classification Maps are collected in Table 2-11.

Table 2-10. Comparisons of GTM and Supervised Classification Maps

Classification Map	Similarity Measures w.r.t. GTM (%)		
	Normal	Forests Merged	Boundary Errors Ignored
DSCM	63.65	66.53	84.89
MLCM	68.41	71.65	88.86
ETCM	67.22	70.25	88.06
LCM	69.25	73.35	88.66
MLMCM	69.16	72.55	90.31
LMCM	72.43	76.19	91.37

Note: All the classification maps except ETCM have six classes. There are seven classes in ETCM, the seventh class resulting from thresholding.

Table 2-11. Comparisons of GTM and Unsupervised Classification Maps

Classification Map	Similarity Measures w.r.t. GTM (%)		
	Normal	Forests Merged	Boundary Errors Ignored
BCM (11)[+]	54.79 (8) 73.59*	76.18 (6)	90.05 (6)
SSCPCM (6)[+]	50.64 (6) 66.64*	65.45 (6)	82.24 (6)
HINDUCM (9)[+]	52.18 (8) 64.43*	82.73 (6)	95.14 (6)

Notes: + The numbers in parentheses indicate the number of classes in the original maps. The numbers in parentheses beside the similarity measures are the number of classes to which the unsupervised classifications were assigned to maximize the similarity measures.

* These similarity measures were obtained by treating as correct the classifications of "Mixed Forest" samples in the GTM as either Deciduous or Evergreen Forest.

2.4 COMPARISONS BETWEEN CLASSIFICATION MAPS

It is seen from Subsection 2.3 that the similarity measures between the ground truth map and the classification maps are all of the same order. This indicates that the classification maps themselves might be quite similar to each other. It is useful to compare classification maps with each other to find how and where they are different. A high similarity measure between the classification maps obtained by different methods but a low one between classification maps and the GTM could cast a doubt on the correctness of the GTM.

Three similarity measures were used in the comparisons, as in the Ground Truth Map cases.

The "normal" similarity measure is simply given by

$$\sum_{i=1}^{6} a_{ii} \Big/ \sum_{i=1}^{6} \sum_{j=1}^{6} a_{ij}$$

The similarity measure when forest classes are merged is given by

$$\frac{\sum_{i=1}^{6} a_{ii} + a_{45} + a_{54}}{\sum_{i=1}^{6} \sum_{j=1}^{6} a_{ij}}$$

The third similarity measure is again determined by ignoring the boundary errors, the boundary points being defined by examining the first map (e.g., MLCM when comparing MLCM versus MLMCM).

Some illustrative similarity matrices and dissimilarity maps are presented that typify the general trend in these comparisons.

Table 2-12 shows the similarity matrix between the supervised Maximum Likelihood and Linear Sequential Classifiers. The concentration of large populations on the diagonal elements of the matrix indicates the close agreement between the classification results. Notable areas of disagreement are between Urban and Agriculture, Transporation and Deciduous Forest, and Deciduous and Evergreen Forests.

The similarity measures in this comparison are

Normal	85.59 percent
Forests Merged	91.04 percent
Boundary Errors Ignored	98.02 percent

Thus clearly the areas of disagreement are primarily due to differences in classifying the colored spectral signatures associated with boundary points.

What is surprising is that the single season Maximum Likelihood Classification agrees less with the Multitemporal Maximum Likelihood Classification than with the Linear Classification. The similarity matrix is shown in Table 2-13 and the dissimilarity map on Figure 2-15. Areas of disagreement are similar to those of the linear case, but the extent of divergence is considerably greater. However inspection of Figure 2-15 shows the majority of dissimilarity points to lie on boundaries. This suggests that registration of the multiple bands of imagery was insufficiently precise, resulting in a "smearing" of the spectral signatures at boundary points. The similarity measures in this case are

Normal	75.92 percent
Forests Merged	79.60 percent
Boundary Errors Ignored	97.76 percent

Considering that the addition of multiple season data improved similarity with GTM by only a few percent at most, the 76 percent normal similarity here and 98 percent boundary suppressed similarities point to the requirement to improve image congruencing techniques.

By contrast the similarity comparison between Maximum Likelihood and ELLTAB classifications, Table 2-14 shows complete agreement, disregarding the few hundred thresholded samples. Figure 2-16 shows the majority of these to be at interior points, as is confirmed by the small increase in similarity when the boundary points are suppressed.

Normal	97.63 percent
Forests Merged	97.63 percent
Boundary Errors Ignored	99.52 percent

As a final illustrative example, the similarity matrix between the Supervised Linear Classifier and the Unsupervised HINDU Classifier is shown, in Table 2-15. Here notable points of similarity include the association of 12123 samples in Cluster 1 with the Deciduous Forest Class 4, 2684 samples in Cluster 2 with the Transportation Class 2, though the large number of samples in this cluster associated with Urban and Deciduous Forest further emphasize the weak spectral structure in the original data. After reassignment of classes to enable more meaningful comparison, the similarity measures become

Normal	71.80 percent
Forests Merged	80.46 percent
Boundary Errors Ignored	99.26 percent

A summary of the complete comparison between the techniques employed is given in Tables 2-16 and 2-17, the former referring to supervised techniques only, and the latter to comparison of supervised and unsupervised techniques.

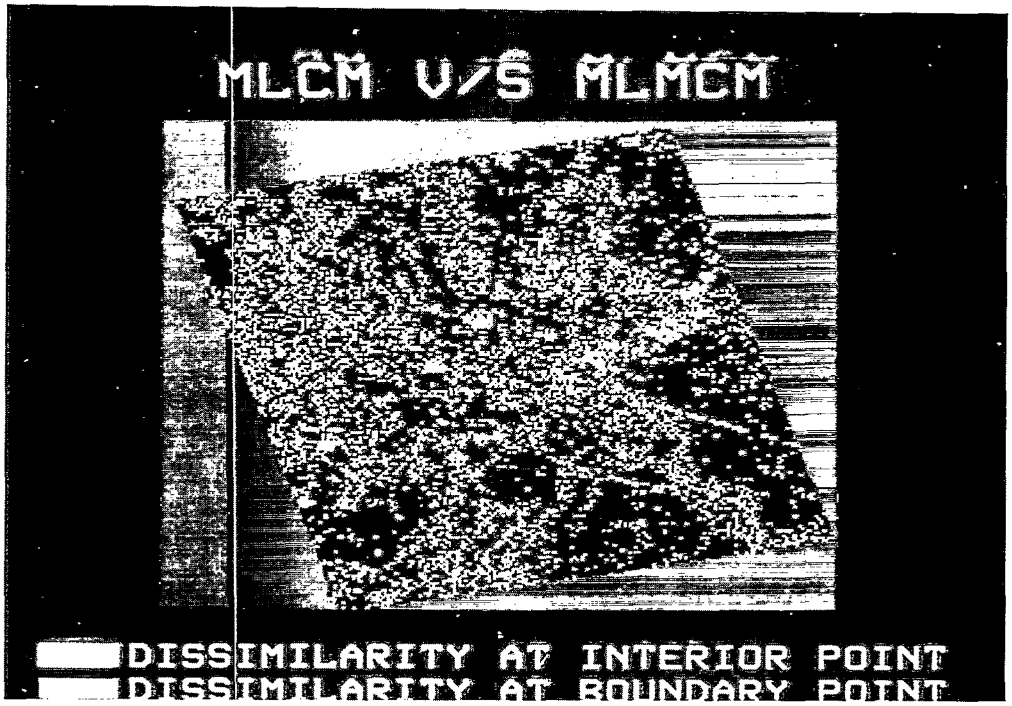

MLCM V/S MLMCM
DISSIMILARITY AT INTERIOR POINT
DISSIMILARITY AT BOUNDARY POINT

Figure 2-15. Dissimilarity Map Between Single Season and Multitemporal Maximum Likelihood Classification Maps

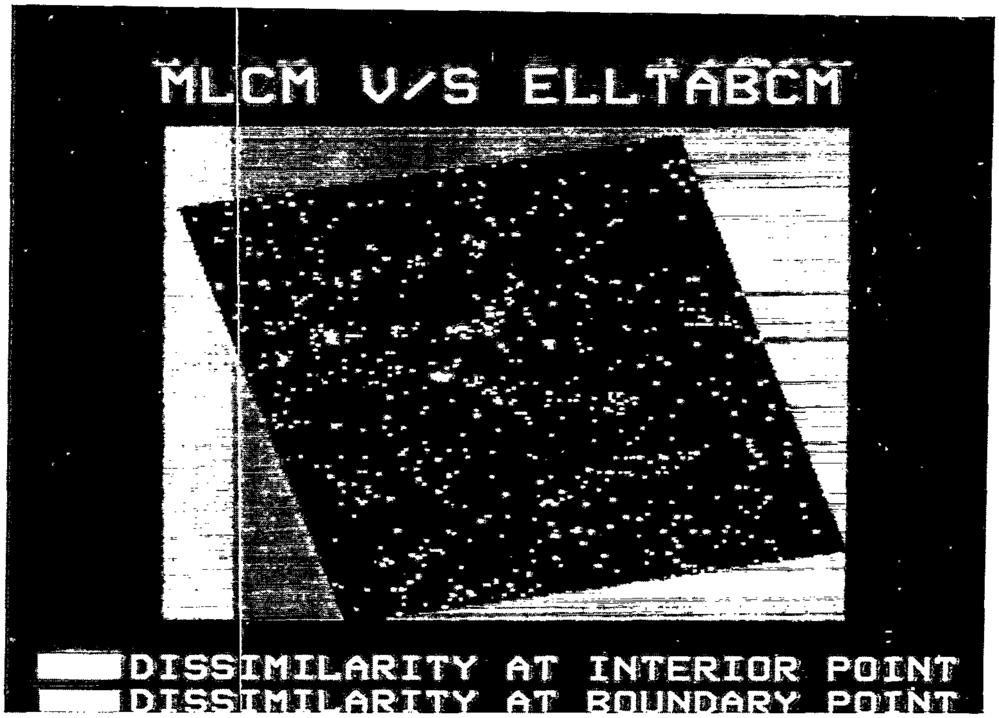

MLCM V/S ELLTABCM
DISSIMILARITY AT INTERIOR POINT
DISSIMILARITY AT BOUNDARY POINT

Figure 2-16. Dissimilarity Map Between Maximum Likelihood and ELLTAB Classification Maps

Table 2-12. Similarity Matrix Between Maximum Likelihood and Linear Classifiers

CLASS NO.	1	2	3	4	5	6
1	2537	108	853	165	26	0
2	160	2482	0	644	164	17
3	212	1	6574	28	22	0
4	28	452	0	14722	844	0
5	0	0	0	972	1816	0
6	0	3	0	10	105	457

MLCM V/S LCM

Table 2-13. Similarity Matrix Between Maximum Likelihood and Multitemporal Maximum Likelihood Classifiers

CLASS NO.	1	2	3	4	5	6
1	1301	786	1562	38	0	2
2	596	1760	153	824	78	56
3	440	66	6330	1	0	0
4	182	1174	39	13667	856	128
5	69	462	5	405	1789	58
6	2	34	2	16	8	513

MLCM V/S MLMCM

Table 2-14. Similarity Matrix Between Maximum Likelihood and ELLTAB Classifications

CLASS NO.	1	2	3	4	5	6	7
1	3492	1	0	0	0	0	196
2	0	3380	0	0	0	0	87
3	0	0	6696	0	0	0	141
4	0	0	0	15783	0	0	263
5	0	0	0	0	2718	0	70
6	0	0	0	0	0	541	34

MLCM V/S ETCM

Table 2-15. Similarity Matrix Between Supervised Linear and Unsupervised HINDU Classifiers

CLASS NO.	1	2	3	4	5	6	7	8	9
1	0	1383	758	0	515	252	29	0	0
2	190	2707	0	0	1	144	0	0	4
3	0	231	4181	2074	617	44	280	0	0
4	11631	4831	51	3	0	9	0	10	6
5	2514	272	17	10	0	0	0	147	17
6	73	2	0	0	0	1	0	220	178

LCM V/S HINDUCM

Table 2-16. Comparisons Between Supervised Classified Maps

Classification Map		Similarity Measures (%)		
Map 1	Map 2	Normal	Forests Merged	Boundary Errors Ignored
MLCM	DSCM	79.37	83.40	96.48
MLCM	ETCM	97.63	97.63	99.52
MLCM	LCM	85.59	91.04	98.02
MLCM	MLMCM	75.92	79.60	97.76
LCM	DSCM	79.94	83.08	97.06
LCM	LMCM	79.30	84.80	98.82
MLMCM	LMCM	78.30	83.35	97.97

Table 2-17. Comparisons Between Supervised and Unsupervised Classification Maps

Classification Map		Similarity Measures (%)		
Map 1	Map 2	Normal	Forests Merged	Boundary Errors Ignored
MLCM	BCM	62.50 (6)	68.00 (5)	96.05 (5)
MLCM	SSCPCM	55.90 (6)	65.18 (5)	91.98 (5)
MLCM	HINDUCM	70.18 (6)	79.09 (5)	99.45 (5)
LCM	BCM	63.37 (6)	68.49 (5)	95.37 (5)
LCM	SSCPCM	56.91 (6)	65.78 (5)	91.12 (5)
LCM	HINDUCM	71.80 (6)	80.46 (5)	99.26 (5)

Notes: The similarity measures reflect the best that can be obtained by a reassignment of the classes in the unsupervised maps to classes in the respective supervised maps.

The numbers in parentheses are the numbers of classes in the Unsupervised Maps (Map 2)

2.5 INVENTORY COMPARISONS

In many instances, the users are simply interested in the inventories or the estimates of percentage occupancies of the various classes over a given region, rather than the point-by-point occurrences of the classes. It is reasonable to expect that the accuracy of the inventories derived from any classification method should be greater than the point-by-point accuracy of the corresponding classification map. This contention is easily justified for a two-class map. Suppose there are M and N points in classes A and B, respectively, and a classifier assigns m and n points from A and B into B and A, respectively. Then the map inaccuracy is $(m+n)/(M+N)$ whereas the inventory inaccuracy is only $|m-n|/(M+N)$.

To explore this for the various classification maps, a study was made comparing the inventory obtained from the GTM (of the Bald Knob Quadrangle) with that from each of the classification maps. Since, in the inventory, it is impossible to account for the "mixed forest" class in the classification maps, the forest classes were merged in the GTM and all the maps. The similarity measure between inventories is defined as

$$\text{ISM} = \left[1 - \sum_{i=1}^{m} |p_{1i} - p_{2i}| / 2 \sum_{i=1}^{m} p_{1i}\right] 100\%$$

where p_{1i} and p_{2i} are the populations of the class i in maps 1 and 2 and m is the number of classes. This definition has the significance that the dissimilarity is measured as the norm of the deviation of either of the inventory vectors from their mean. This definition assures that the ISM is between 0 and 100 percent, agreeing with the intuitive concept of similarity. When the inventories are identical, the ISM is 100 percent and when they are most dissimilar (with all samples assigned to one class in map 1 and to a different class in map 2) the ISM is 0. The factor 2 in the denominator is used to assure this.

The results of this experiment are shown in Table 2-18 as S_3. Also shown are the similarity measures S_1 and S_2 corresponding to the pixel-by-pixel comparison with the forest classes merged and with the boundary errors ignored. (These are repeated from Tables 2-10 and 2-11.) In the case of the unsupervised classification maps, the assignments used to compute S_3 were the same as for S_2.

Table 2-18. Inventory Similarities Relative to GTM

Map	Similarity Measures (%)			S_2-S_3
	Pixel by Pixel (Forests Merged) S_1	Boundary Errors Ignored S_2	Inventory S_3	
DSCM	66.53	84.89	77.46	7.43
MLCM	71.65	88.86	81.94	6.92
ETCM	70.25	88.06	81.94	6.47
LCM	73.35	88.66	85.46	3.20
MLMCM	72.55	90.31	82.56	7.75
LMCM	76.19	91.37	88.01	3.36
BCM	76.18	90.05	90.08	-0.03
SSCPCM	65.45	82.24	79.27	2.97
HINDUCM	79.42	92.93	92.70	0.23

Table 2-19. Heuristic Analysis of "Bias" (S_2-S_3)

Map	S_2-S_3 (Nearest Integer)	Components			
		Training Area Selection	Forcing Parametric Distribution	Cluster Merging	Lack of Decision Rigor
BCM	0	–	–	–	–
HINDUCM	+1	–	–	–	1
SSCPCM	3	3	4	-4	–
LCM	3	3	–	–	–
LMCM	3	3	–	–	–
DSCM	7	3	–	–	4
MLCM	7	3	4	–	–
MLMCM	8	3	5	–	–
ETCM	6	3	3	–	–

Figure 2-17 gives a pictorial representation of the values of S_1, S_2, and S_3 for the various classification maps. The "normal" similarity measures from Tables 2-10 and 2-11 were used as the abscissae to mark the various classification maps. The dashed line indicates a separation between unsupervised and supervised methods. As noted in Section III-2.3.1, the "normal" similarity measure in the case of unsupervised methods is computed somewhat differently from that for supervised methods (due to the existence of a mixed forest class on the GTM). The ordinates indicate, for each classification map, the similarity measures S_1, S_2, and S_3 on the curves with the appropriate names. As expected, the inventory similarity measures are greater than S_1. This increase ranges from 10 to 14 percent. In the case of BCM, S_2 and S_3 are approximately equal. This indicates that the compensatory effects of misclassifications by the decision boundaries in feature space in inventory evaluation are comparable to the effects of neglecting misclassifications at the spatial boundaries. A heuristic explanation accounting for the various components of S_2-S_3 is given in Table 2-19. The several biases indicated should only be regarded as a first attempt at identifying the causes for the differences. Notable are the groupings of the three parametric supervised maps (ETCM, MLCM, and MLMCM), the nonparametric maps (LCM and LMCM), and the unsupervised maps (HINDUCM and BCM) which do not use any information on spatial coherence or homogeneity of training sets. While SSCP does use parametric descriptions of the clusters and homogeneous training areas, the biases introduced by these may be looked upon as being counteracted due to the provision to merge the clusters so obtained. The bias introduced in the case of supervised methods can be viewed as that due to choosing a few homogenous training areas and forcing good discrimination of them which does not necessarily imply generalizability to the entire data set. The decision rigor involved in quantifying the feature space into regions of different classes or clusters is least in DSCM and this introduces certain bias. In the case of HINDUCM also there is a certain lack of rigor introduced by the table lookup scheme of classification (being based on use of prototypes for creating the label table) but to a much lesser degree than in DSCM. This area of accounting for the various types of differences between classification maps and quantifying them requires further study with respect to other classification methods and data sets.

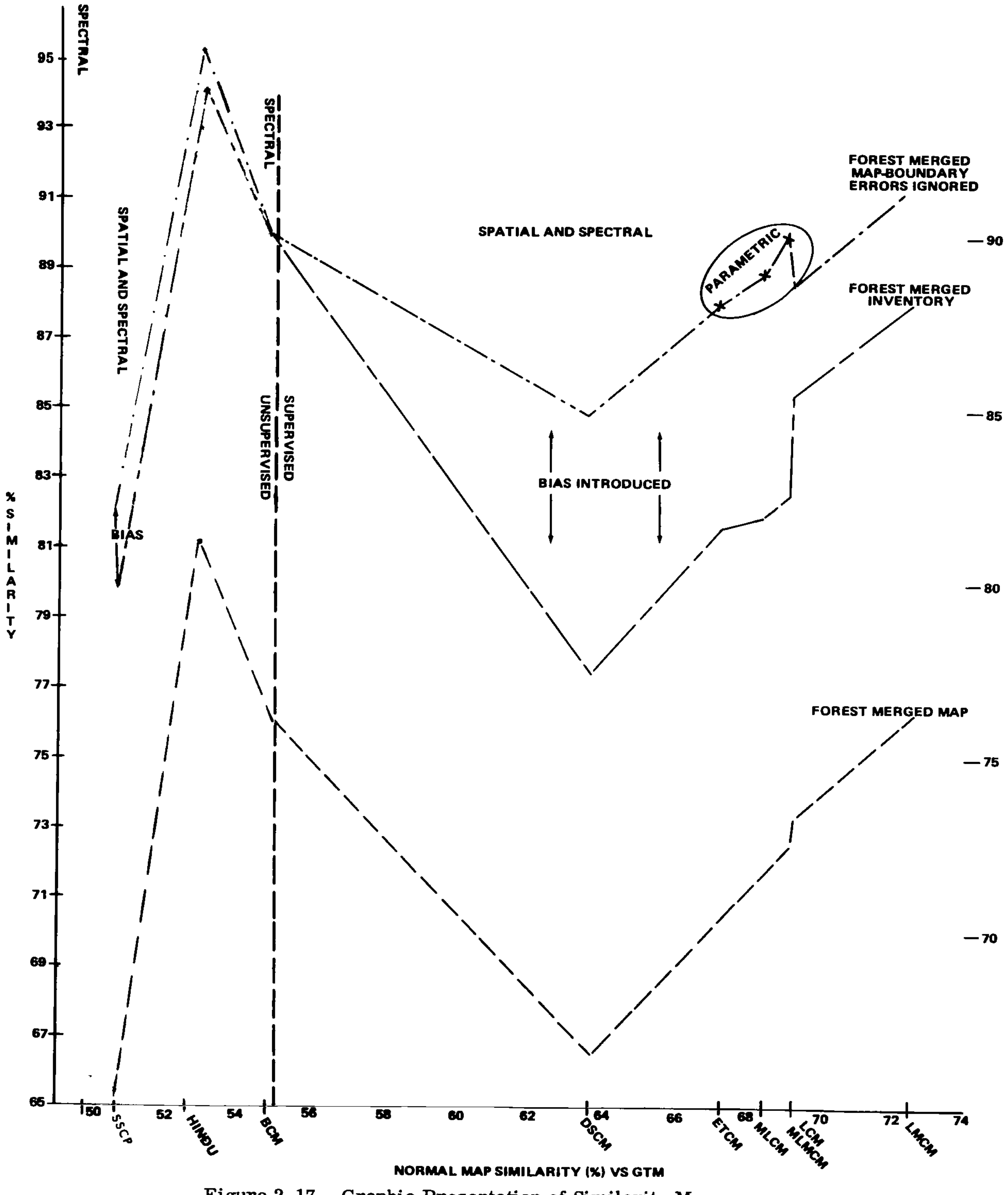


Figure 2-17. Graphic Presentation of Similarity Measures

2.6 USER ASSESSMENT

The following response to the user questionnaire described in Section I-4.4 was received from Mr. John Wilson, Director of Natural Resources, Tennessee State Planning Office, Nashville, Tennessee.

TENNESSEE

STATE PLANNING OFFICE

RAY BLANTON
Governor

NILES SCHOENING
Director

660 CAPITOL HILL BUILDING
301 SEVENTH AVENUE, NORTH
NASHVILLE, TENNESSEE 37219
615-741-1676

December 11, 1975

Mr. Robert Jayroe
Marshall Space Flight Center
Marshall Space Flight Center, Alabama

Dear Bob:

In response to the user evaluation questionnaire, the following answers have been prepared:

Question 1 - How well was the computer analyses able to satisfy the land use requirements?

In general, I would have to say that the computer analyses were able to satisfy our requirements, but certain qualifications need to be discussed. First, the accuracy figures derived from directly comparing the classifications maps with the ground truth map (normal column) are surprisingly low. Part of this problem may be due to the fact that the training and analysis was limited to a small area (52,000 pixels) and obtaining pure features for training were difficult. More accurate results probably could have been obtained if the entire image had been used as a resource for training. Secondly, the accuracy assessment pointed out that pixel mixing, at the boundary between two different features, was the dominant problem. More resolution in the data may help this problem, but in order to understand the mixing problem, ground checks need to be made of the classification map data in question.

Question 2 - Which computer technique best satisfies the land use requirements?

Based upon the accuracy assessments, the answer would have to be the multi-temporal linear classifier. However, the accuracy assessment also indicates that there is no technique that is outstandingly superior to all of the others.

Question 3 - What cost/benefit, if any, would you derive from using computer versus conventional photo-interpretive techniques?

The computer associated costs are approximatly $800 for data tapes (4 temporally different tapes at $200 each) and roughly $200 for the analysis, which totals $1,000. The cost for the conventional data and phto interpretation is approximately $1,300 ($1,000 for the orthophotoquad, $300 for analysis and cartography). The computer analysis would appear to be more cost/effective, especially on larger areas, since there is an initial cost that is not area dependent (cost of original data and training) and the classification area analysis, once the training has been completed, is a small fraction of the photointerpretive costs. The cost of photointerpretation, however, is consistantly proportional to the size of the area. There is a need to qualify this discussion from two points of view. First, I am referring to the classification of gross features only (agriculture, forest, urban, etc.); and secondly, our evaluation is an experiment concerned with using high altitude photography, which should not be confused with the more conventional and operational procedures that have been used in the past.

Question 4 - What improvements or changes need to be made, if any, in the computer analyses?

First,I think that areas other than the original test site should be included for selecting training data and analyses, even though the accuracy assessment is to be performed on the test site only. Secondly, there is the need to supply the user with geographically correct photoproducts that is appropriately scaled with the originally supplied ground truth map. Thirdly, the geographically-corrected classification maps need to be studied to understand the pixel mixing problem and improve the classification techniques. This study should include extensive field work.

Question 5 - What do we consider to be the short-comings and good points of each technique result?

Generally speaking, the unsupervised techniques appear to lack the ability to pick out the detail that can be achieved with the supervised techniques, and this appears to be mainly a resolution problem. We will be better able to answer this question when we receive the geometrically corrected classification maps.

Question 6 - How would we rank the technique results in order of satisfying the land use requirements?

A visual comparison of the maps as to priority by accuracy resulted in the following list:

1) Multitemporal ML class
2) Maximum Likelihood class
3) Multitemporal Linear class
4) Linear class
5) Elltab class
6) Hindu cluster
7) Density Slicing class
8) Spatial and Spectral class

Mr. Robert Jayroe
Page 3
December 11, 1975

Question 7 - Which, if any, of the techniques provide information beyond what is contained in the ground truth maps, that is useful or improves the accuracy? If so, what new information was provided or where was the improvement noted?

The most significant improvement was the ability to obtain a numerical percentage of the features contained in the map. Further examination of the classification maps presently indicate that the maps provided no additional new information, and in some cases less, when compared to the ground truth map; but a geographic corrected version of the classification maps is needed to confirm this observation. As an example, it was not possible to delineate the TVA power lines, unless the June data was used by itself or in a multitemporal set. Again these statements have to be qualified in that we are only concerned with gross features where new types of information are not too likely to occur in this particular area of the data sets.

Question 8 - What would be your opinion on using the best technique at your facility, at your cost to produce computer land cover maps for public use?

From the cost/effective and operational point of view previously discussed the suggestion appears attractive, but I am concerned that the initial cost and continued resource support may be prohibitive. As you know we have participated in this endeavor only in the initial phase of identifying a specific feature, and in evaluating the results. I feel that we could better understand the effort in its entirety and the resources involved and state a more specific opinion if we had had the opportunity to participate in the analysis directly.

If you need further information, please let me know.

Sincerely,

John M. Wilson
Director, Natural Resources Section

2.7 OBSERVATIONS AND CONCLUSIONS

Certain observations and conclusions will be drawn from the study of this data set and it is, therefore, extremely important that the following statements not be misunderstood, quoted out of context, or generalized to all possible technique analysis results. The observations and conclusions represent only a beginning in the study of classification technique assessment and at present have to be qualified by the following statements:

Qualifying Statements

- The present results are based only on one data set, the Bald Knob, Tennessee, Quadrangle. Thus, some observations and conclusions are probably data set dependent.
- The present results consider only one application; namely, land cover mapping. Thus, some observations and conclusions are also probably application dependent.
- The computer analysis was performed on Landsat data, while the ground truth information was obtained from higher resolution aerial photography and photointerpretation. Thus, a comparison between apples and oranges may have occurred in some cases, although the intent was honorable.
- Possible inaccuracies and insufficient detail in the Ground Truth Map make the automatic classification map results appear less accurate.

With respect to the Ground Truth Map, no error analysis is available concerning the differences that would occur had the Ground Truth Map been developed by several different photointerpreters, or had the ground truth map been developed from Landsat imagery rather than from aerial photography. In either case, the differences in observation would probably have occurred mainly in the urban category, where context is used in the photointerpretation decision process, and in the mixed forest (deciduous and coniferous) category where photointerpretation is not necessarily precise. Attempts at classifying a mixed forest category with computer analysis were unsuccessful as expected, since mixed implied an aggregation of pixels and the computer classification techniques consider individual pixels. The result was that pixels contained in the mixed category were usually classified either as deciduous or coniferous.

The fact that computer techniques deal with individual pixels is illustrated by the frequent incidence of isolated points in the classification maps exhibited in Figures 2-4 through 2-13. Comparing these with the Ground Truth Map, Figure 2-3, the latter is seen to have much greater homogeneity within classes. Since no field survey was conducted in preparing the Ground Truth Map, it is not possible to assert that all individual point differences are classification errors, though the similarity measures are reduced by this effect.

- The small scale of the test site and its ground features makes selection of pure data samples for training the supervised classifiers very difficult, so degrading their accuracy and biasing their results.

The majority of the errors in misclassification occurred at the boundaries between two or more different features, which will be discussed later, and in not satisfactorily distinguishing the urban from the agriculture category. Part of the urban/agriculture confusion is due to the fact that no pure urban area existed in the Landsat data for the test site, and therefore the training areas for urban, which were widely scattered houses in a rural community, unavoidably contained data relating to agriculture. Also the areas designated as urban on the ground truth map were derived from the higher resolution aircraft photography where individual houses could be seen and included surrounding areas which were agricultural. If one considers classifying individual pixels of Landsat data without the higher resolution aircraft imagery providing contextual information, then there probably is no discernible urban category in the Landsat data test site because of insufficient resolution. This statement is supported by the fact that the areas of urban/agriculture confusion are high reflectance areas in the Landsat data and visually the classification maps agree more with the Landsat imagery than the aircraft imagery.

- Classification errors appear to be caused mainly by pixel mixing, due to insufficient spatial resolution in the data, and this effect is predominant at class boundaries. Geometric errors also result in apparent classification errors.

Examination of difference maps between the classification results and the Ground Truth Maps reveal, as previously mentioned, that the majority of errors occurred at the boundaries between two or more different features. Based upon this comparison, there are several possible sources of errors that can be mentioned although their individual contributions to the total error have not been determined. These sources of errors could have resulted from:

- photo-misinterpretation, leading to inaccuracies or insufficient detail in the Ground Truth Map,

- imprecise location of boundaries between features on the original Ground Truth Map,

- imprecise location of boundaries between features on the Ground Truth Map that had to be duplicated from the original for photographing and digitizing,

- imprecise registration in matching geographic coordinates of the digital Ground Truth Map with pixel coordinates of Landsat data, and

- pixel mixing in both digital Ground Truth Map and Landsat data.

An additional possible source of error contributed by the analysis was in the selection of training areas, which in some cases were necessarily diminutive and irregular in size and small in number of pixels for several features. However, it appears that the dominant error is a result of pixel mixing, since all of the classification maps agree with each other more than with the ground truth map, and difference maps between different classification results still indicate that the majority of disagreements occur at the boundaries between two or more different features.

- Comparative classification accuracy between the various techniques improves significantly if classes are homogeneous and if boundary effects are suppressed.

The accuracy assessment permits one to draw mixed conclusions about the accuracy of supervised versus unsupervised techniques, but it is suspected that these conclusions are mainly due to the nature of the data set alone. The data set is predominantly agriculture and forest with the other features occurring in scattered narrow or small patterns that represent a much smaller portion of the data set, and use of the similarity tables results in the unsupervised technique maps being compressed mainly into the forest and agriculture categories. The reason for this is that SSCP requires at least a 7 by 7 pixel homogeneous area in which to cluster and in the case of HINDU the unearthing of a class is highly dependent on the class population and distribution dispersion in the data set. Hence, the unsupervised techniques had little or no chance of discovering and distinguishing the smaller features that were present in the data set. The above statements are supported by examining the similarity measures for three different cases.

If one compares the classification maps with the Ground Truth Maps, the highest similarity for the six class supervised technique is 72 percent, while the highest similarity for the eight class unsupervised technique is only 55 percent. The percentages appear rather low and may be somewhat misleading since approximately 30 percent of the data set are boundary pixels which account for most of the error. In this comparison, it was assumed that the boundary pixels on the Ground Truth Map were correct.

In the second case, if one considers deciduous and coniferous as only one category, forest, the highest similarity measure increases to 76 percent for the six class supervised technique and to 79 percent for the six class unsupervised technique. The unsupervised techniques are about 3 percent more accurate than the supervised techniques, but mainly because the data set is predominantly agriculture/forest.

In the third case, if one further considers that the boundary pixels are classified correctly by each technique, then the highest similarity measure increases to 91 percent for the six class supervised technique and to 93 percent for the two class unsupervised technique. Again, the predominance of agriculture/forest in the data set gives the unsupervised techniques a slight edge in accuracy. It is interesting to note that density slicing is only about 5-10 percent less accurate than the other supervised techniques on this particular data set for land use application, but generalization of this statement to all cases is considered rather risky. It may also be argued that the particular 5-10 percent increase in accuracy may be the most significant 5-10 percent.

- Use of Landsat data from more than one season had a negligible effect on classification accuracy for this particular test site.

With regard to using multitemporal data to increase the accuracy, it was found for this data set and application the similarity increased by 3 percent at the most. Again a generalization is rather risky because the accuracy increase will be highly dependent upon the test site as well as application. It is also interesting to note that the method used for band or channel selection in the multitemporal linear classification and density slicing techniques picked the infrared band (.8-1.1 microns) for classifying all of the features except deciduous and coniferous. The .6-.7 micron and .7-.8 micron bands were chosen to distinguish these two features and the .5-.6 micron band was a last choice in all cases. The same choices were apparent in the multitemporal data with the addition that the winter scene was relegated to a last choice.

- The various techniques tested are comparable in accuracy and, in general, differences between results lie within a spread of about 10 percent.

Examination of the worst case similarity measures for this data set and application indicate that there is less than 10 percent difference between the various supervised technique results under all cases examined. The maximum difference between the various unsupervised results was 15 percent, 10 percent spread being a more typical value. There is no more than 16 percent difference between the unsupervised and supervised technique results. If the classes deciduous and coniferous are considered as one class, forest, and if pixel mising is ignored by assuming that each technique classifies boundary pixels correctly, then the difference between all of the techniques is less than 10 percent.

- Insufficient spatial resolution in the data and pixel mixing are the chief problems in reliable computer classification of Landsat data for inhomogeneous rural areas. Without these problems, over 90 percent classification accuracy is attainable.

The two main problems appear to be lack of resolution in the satellite data and pixel mixing. Since the unsupervised techniques have more difficulty in distinguishing the smaller features, it would appear that increased resolution would provide them with the most improvement. In any event increased resolution would probably increase the accuracy since the number of boundary pixels would tend to increase at a linear rate compared to an area rate for the number of pixels adjoining a boundary. However, an increase in resolution would not necessarily improve the pixel mixing problem and in some cases may actually magnify the problem. One possible approach to handling the pixel mixing problem is to employ boundary detection or enhancement techniques to first separate the data into homogeneous areas, which have a minimum of pixel mixing, and boundary pixels. Once the boundary pixels have been isolated from the rest of the data, special techniques can be developed for handling them in a more effective manner.

3.0 PERFORMANCE SUMMARY AND RECOMMENDATIONS

3.1 PERFORMANCE SUMMARY

For ease of reference, the classification techniques described in Section II and assessed in Section III-2 are briefly summarized. Table 3-1 identifies the chief characteristics of each technique considered. Table 3-2 summarizes the cost and performance data acquired in test and evaluation. Figure 3-1 protrays graphically the percentage similarity measures between a Ground Truth Map and the various classifier results for the following cases.

Normal — pixel-by-pixel comparison with Ground Truth.

Forest Merged — Same as Normal, but with a mixture of two Ground Truth classes considered as one.

Boundary Error Ignored — Same as Forest Merged, but with all pixels on the boundaries between classes being disregarded.

Forest Merged Inventory —Similarity based on comparison of class populations without regard to pixel location.

Table 3-1. Characteristics of Techniques Assessed

Classification Method	Training Data Required	Data Distributions or Statistics Required	Comments
Density Slicing	Yes	No	Classification decision based on tonal differences within a single spectral band. Threshold values for decision rule determined by visual experiment or using a feature selection technique.
Maximum Likelihood	Yes	Yes	Assumes data have Gaussian probability distributions. Gaussian parameters (mean and covariance) for each class estimated from set of known samples. Class assignment made by computing largest decision function, using class Gaussian parameters.
ELLTAB	Yes	Yes	Partitions measurement space into regions associated with each class, determined by analyzing known samples to find their Gaussian distribution properties. Each region is defined in a table. Class assignment made by table lookup.
Linear Sequential	Yes	No	Separates classes sequentially in order of separability by linear surfaces (hyperplanes). These surfaces are determined by iterative analysis of samples whose identify is known.
Binary	No	No	Determines regions of data homogeneity (clusters) by associating the relative magnitudes of spectral band reflectances with a binary decision vector.
Spatial and Spectral Clustering (SSCP)	Optional	Calculated Internally	Identifies regions of spatial homogeneity by mapping boundaries. Combines boundary map with original data to define spectrally similar clusters. Class assignment is by a minimum distance decision rule.
HINDU	No	No	Estimates the local density of samples in measurement space by computing histograms, and automatically locates centroids of data clusters. Class assignment is by table lookup of labels corresponding to the Centroids of the occupied histogram cells, with the table being created using a piecewise linear discriminant classifier.

Table 3-2. Cost/Performance Summary

Technique	# of Channels	# of Training Areas	Training Time (Sec)	# of Pixels	Pixels Per Sec	# of Classes	Classification Time (Sec)	# of Pixels	Pixels Per Sec	Computer Cost $ (3)	Manpower Cost $ (1) (4)	Total Cost $
Supervised			(1)									
Density Slicing	4	36	17	-	-	6	16.5	52,000	3152	3.26	160	163.26
Maximum Likelihood	4	36	4.5	-	-	6	114	52,000	457	11.52	160	171.52
Multitemporal Maximum Likelihood	16	36	12	-	-	6	690	52,000	75	68.25	160	228.25
ELLTAB	4	36	4.5	-	-	6	63 (2)	52,000	825(2)	6.56	160	166.56
Linear Sequential	4	36	40	-	-	6	18	52,000	2970	5.64	160	161.64
Multitemporal Linear Sequential	16	36	900	-	-	6	39	52,000	1337	91.29	160	251.29
Unsupervised		# of Clusters										
Binary Classifier	4	-	-	-	-	11	499	230,400	482	48.51	-	48.51
Spatial & Spectral Clustering	4	58	1208	60,000	50	5	751	60,000	80	190.46	N.A.	190.46(5)
HINDU	4	9	24.5	52,000	2080	9	0.60	52,000	8666	2.45	N.A.	2.45 (5)

Notes: (1) The Supervised Techniques required two man-days analyst time to select and refine training samples.
(2) UNIVAC 1108 time (all others apply to IBM 360/65)
(3) $350/Hour
(4) $10/Hour
(5) Does not include analyst time, which cannot yet be estimated for operational work.

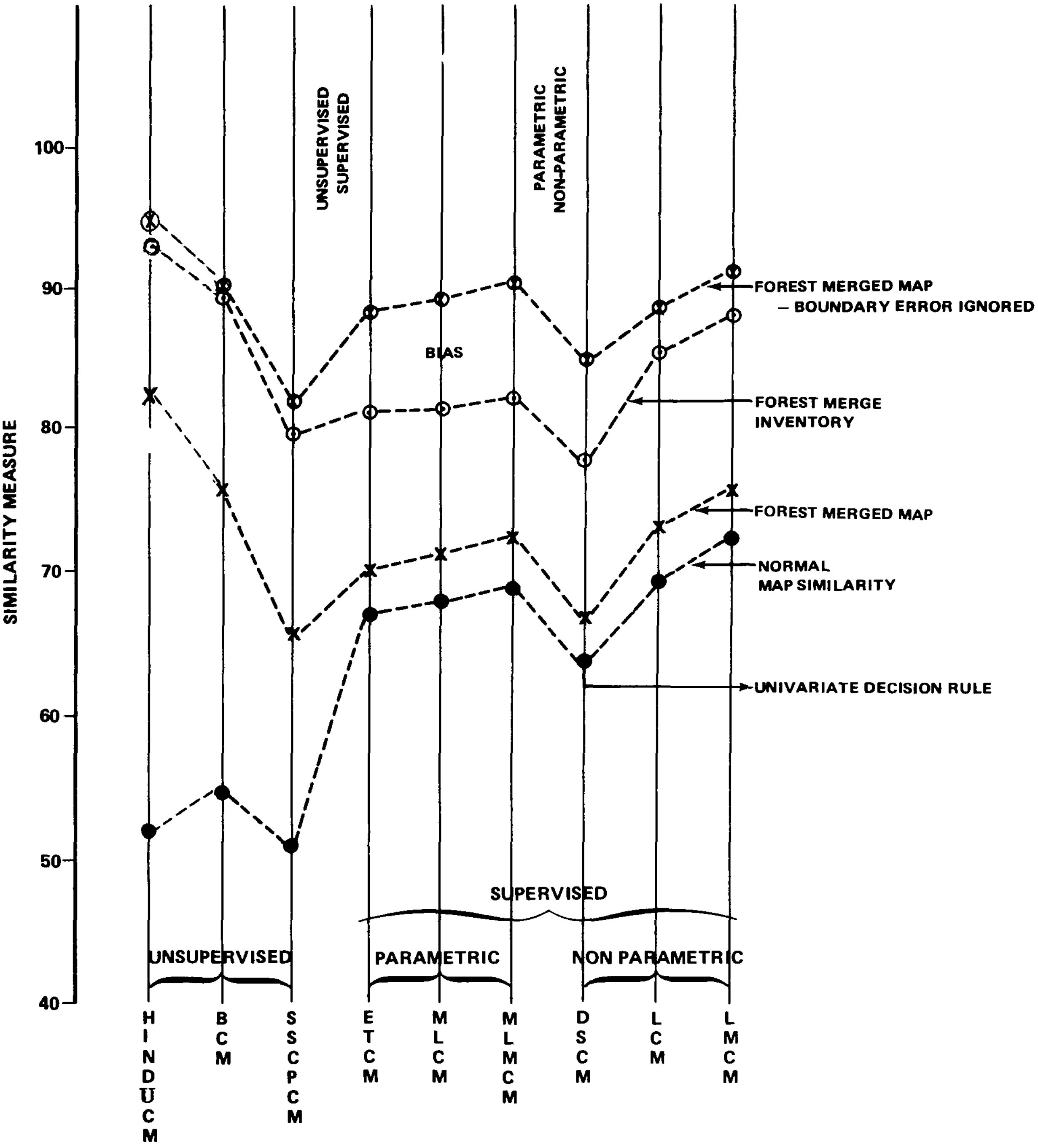


Figure 3-1. Summary of Similarity Measures Between the Classification Maps and a Ground Truth Map

3.2 RECOMMENDATIONS

Based upon the results of this initial study, there are several areas of activity that need to be further pursued in order to obtain more visibility on existing techniques via assessments. These activities include:

- Problem separation —Determine and isolate those problems that can be attributed to sensor specification only, data sets only, techniques only, and applications only.

- Problem synthesis — Determine the overall dependence of the above-mentioned independent problem areas.

- Develop methods to properly understand and handle the pixel mixing problem.

- Extend the results of signature extension in a geographical and temporal sense.

- Extend the results to other temporal, spatial, and spectral domains to determine the time frames, resolution, and spectral bands and bandwidths needed for a particular discipline application.

- Extend the results to other discipline applications to determine whether one technique is more suitable to a particular application than another.

- Extend the results to include multilevel data bases.

If the above problems can be effectively solved, then optimum impacts can continue to be made of future operational remote sensing platforms and efforts can be initiated to provide more cost/effective and timely automatic processing.

> *"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."*
>
> —NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546